

BSD *on* Windows 1.5

アスキー出版局

BSD on Windows 1.5

USERS MANUAL

ユーザーズマニュアル

アスキー出版局

商　標

はじめに

Windowsを使っていますか？ Windowsに使われていますか？ Windowsを使用していてふと気が付くと、同じ操作を繰り返し実行していることがよくあります。楽をするためにコンピュータを使っているはずが、少しも楽になっていないのではないでしょうか。Windowsが提供する環境は、あらかじめプログラミングされた定型処理なら、簡単な操作で実行することができます。しかし、それから外れたことを行なう場合は、同じ操作を繰り返さなければならない場面に数多く出くわします。

一方、BSDが提供する環境は、Windowsのプログラムとは対照的に不親切です。BSDが提供する環境は完成品ではありません。ユーザーが、ツールを組み合わせて問題を解決するために必要になるな機能を提供する、半完成品です。ほかのプログラムと組み合わせて使うときにその真価を発揮します。しかしWindowsが苦手とする不定形の処理もツールの組み合わせで行なうことが可能です。

筆者がWindowsに移行しようとした際に、直面した問題は、使い慣れたエディタ(emacs)がないということもさることながら、不定形処理を解決するためにBSDが提供しているツールキットがないということでした。BOWはそれを解決するために開発しました。

BOWはWindows上にBSD互換環境を構築することを目的に開発した、BSDカーネルエミュレータです。システム管理に使用する一部のプログラムを除く殆どのBSDプログラムを、本物のBSDのように組み合わせて実行することができます。

BSDが長期間かけて成熟してきたコマンドラインインターフェースと、GUIでのイージーオペレーションを融合するような環境を構築する一助になれば幸いです。

GNUを支援する方々、BSDを開発したCSRGのスタッフを始めとするBSD hackersに感謝します。彼らの仕事がなかったならば、BOWは生まれませんでした。

BOW 1.0のユーザーの皆さん、富樫さんを始めとするベータテスターの皆さんには、貴重なご意見をいただきました。スーパーアスキー編集部のスタッフは本パッケージの完成と出版に終始支援下さいました。この場を借りて深くお礼申し上げます。最後に妻の千湖と息子の海渡に感謝します、家族の理解がなかったならば、日曜プログラマは本パッケージを仕上げることはできなかったでしょう。:-)

1996年3月30日

太田博志

マニュアルの読み方

本マニュアルは、Windows(Windows 3.1 または Windows 95)の基本的な操作と、UNIX (BSD) の個人環境やアプリケーション環境の設定を自力でできるユーザーを対象に構成されています。Windows の各種の設定ファイルや、BSD の.login、.cshrc、.profile、.emacs などの内容をある程度理解でき、必要ならば自力で修正できるユーザーが対象となります。Windows や BSD に関する用語でわからないものが出てきたら、それぞれのマニュアルや関連書籍を参照してください。BSD のコマンドに関するドキュメント類は、原則として「BSD プログラム CD-ROM」に収めたソースファイルなどに、テキストファイルの形式添付されています。また、関連書籍や参考文献は「APPENDIX D　参考文献」にまとめてあります。

技術用語などは基本的に Windows のマニュアルの表記を採用しています。また、UNIX 固有の用語などは、BSD のオンラインマニュアルの表記を参考にしています。

BSD コマンドやシステムコールについては、BSD のオンラインマニュアルのセクション番号を付記しています。たとえば、「cp(1)」という表記は、cp コマンドのオンラインマニュアルがセクション 1 に収録されていることを示します。

セクション番号の意味

(1)	general commands	コマンド
(2)	system calls	システムコール
(3)	libraries	ライブラリ
(4)	devides	特殊ファイルとハードウェア
(5)	file formats	定義ファイルなどのフォーマット
(6)	games	ゲーム
(7)	miscellaneous	その他
(8)	system maintenance commands	メンテナンス

本マニュアルの「5 章　BSD コマンド入門」では、BSD になじみの薄いユーザーの便宜をはかるため、主要な BSD コマンドについて簡単な解説をしています。しかし、各コマンドのオプションや使い方について、網羅的に説明するものではありませんので、より詳しい使い方については、各コマンドのオンラインマニュアルなどを参照してください。

BSDでは、仕様上の制限(または修正する予定のないバグ)を、すべて「Known Bugs」として扱っています。BOWでもこれらの制限は継承されています。また、BOW自体のKnown Bugsについては、「7章　BOWの仕様」に記述されています。ファイルシステムの仕様については、Windows 95またはMS-DOSのファイルシステムの持つ制限が継承されます。

付属のCD-ROMについて

付属のCD-ROMには4.4BSD-LiteをベースにしたBSDのカーネルソース、各種コマンドのソースファイル、バイナリファイルが収録されていますが、これらのプログラムは基本的にすべてフリーソフトウェアです。

CD-ROMに収録されたファイルの再配布(バックアップメディアなどでの譲渡やパソコン通信やインターネットなどへのアップロードなど)は、各ファイルまたはアーカイブごとにそのオリジナルの作者が規定した配布条件に従ってください。

株式会社アスキーは、CD-ROMに収録された内容について使用/配布制限を設けません。また、一切の権利も主張いたしません。したがって、CD-ROMの内容については一切のサポートはできませんので、ご了承ください。また、CD-ROM内のプログラムの作者に対して、BOW上での動作に関するお問い合わせは絶対におやめください。

付属の3.5インチフロッピーディスクについて

付属の3.5インチフロッピーディスクに含まれる内容は、BOWの作者である太田博志氏の著作物として著作権法によって保護されています。一般の市販ソフトウェアと同様に、バックアップ以外のコピーはできませんので、ご注意ください。

詳しくは、フロッピーディスクのパッケージ表面に印刷されている「エンドユーザー使用許諾契約書」をお読みください。フロッピーディスクをパッケージから取り出すと、この契約書に同意したものとみなされますので、ご注意ください。

目次

1章

システムの概要

本章では、BOW のシステムを理解するための基本的な概念や、BOW の持つ機能の解説、BOW を使うために必要なシステム環境などを解説します。BOW を初めて使われる方は、かならずお読みください。また、本章の最後では、「BSD on Windows 1.0」から追加された機能を解説します。

1.1 BOWとは何か

BOW は、Microsoft Windows の 16 ビット API（以下 Win16 API と略す）の上で、4.4BSD 互換の BSD システムコール環境をエミュレートするシステムです。インテル系 CPU で動く BSD（Free BSD、Net BSD など）のバイナリファイルを、そのまま動かすことができます。また、BSD 用のプログラムのソースファイルを BOW 上でコンパイルして使うこともできます。ユーザーインターフェイスとしては、ANSI 互換のキャラクタ型端末のエミュレータである「BOWPAD」を用います*1。

ノンプリエンプティブなマルチタスク API である Win16 API の制限は受けますが、BOW 上のプログラムはマルチタスク／マルチプロセスで動作します。シグナルによるプロセス間通信も可能です。csh などのジョブ制御機能を使えば、複数のプログラムをバックグラウンドで動かしながら、他の操作を行なうこともできます。また、複数の BOWPAD ウィンドウを開いて、複数のプログラムを切り替えながら使うことも可能です。

BSD とバイナリ互換であるため、テキストエディタなどのアプリケーションや、各種コンパイラや make などの開発環境*2、perl、sed、awk などのフィルタ系ツール、各種シェルなどは、一般の BSD システムと同じ物が使えます。

1.1.1 BOWの動作原理

インテル系 CPU 用で動作する BSD システムでは、BSD システムコールは LDT*3の 7 番が示すセグメントのオフセット 0 を far コールすることで実現されています。BOW はこのしくみをそのまま踏襲して、LDT の 7 番に専用のシステムコールハンドラのアドレスを置き、BSD のシステムコールが発生したら、このシステムコールハンドラで処理します。このハンドラによって、各 BSD システムコールは Windows の API へ翻訳されます。最終的に、すべての BSD システムコールは、Windows API のみで処理されます。

システムコールの発行方法が既存の BSD システムと同じなので、BOW 上では基本的にインテル系 CPU の BSD で使われているバイナリファイルを、再コンパイルやファイル

---

*1 X Window などのグラフィックベースのユーザーインターフェイスは使えません。市販されている MS Windows 用の X サーバーソフトウェアと併用することは原理的に可能ですが、動作の保証はいたしません。

*2 gdb などのデバッガは、BOW 上ではプロセッサの発するデバッグ例外などの例外を捕らえることができないため、BOW では動きません。

*3 Local Discpritor Table の略。386 以上のインテル互換 CPU で、保護仮想アドレスモードを実現するために、実アドレスと仮想アドレスの対応づけを行なう対応表の一種。

変換などを行なう必要なく、そのまま動作させることができます。一部のシステムコールについては、動作しないもの、正規のBSDとは異なる動作をするものがありますが、アプリケーションを使用するにあたって、できるだけ影響のないようにインプリメントしてあります。システムコールの互換性については、181ページの表を参照してください。

1.2 BOWにできることとできないこと

前述のように、BOWはBSDをバイナリファイルレベルでエミュレートする環境ですが、ファイルシステムや各種デバイスのサポートについては、Windowsに依存しています。このため、一般のBSDシステムと比べて、機能に制限があったり、動作が異なる場合があります。ここでは、一般のBSDシステムと比較して、BOWでできること、できないことなどを解説します。

1.2.1 BOWとBSDの関係

BOWのシステムは、大きく分けると次の3つの部分から成り立っています。

1. Windows（およびDOS）
2. BOW本体
3. BSDバイナリ群

BOWのパッケージの内容でいえば、2.は「BOWシステムフロッピーディスク」、3.は「BSDプログラムCD-ROM」に相当します。3.の部分については、一般に流通している「Free BSD」などのBSDシステムに含まれるバイナリプログラムとまったく同じ物です。1.と2.の組み合わせが、一般のBSDシステムでいうところの「カーネル」と「デバイスドライバ」に相当します（図1-1）。

一般のBSDシステムでは、システムが起動するとき、ブートローダなどのプログラムによってカーネルとデバイスドライバがロードされ、つぎに各種デーモンなどのBSDプログラムがロードされて、ユーザーがログインできる（ユーザーのプログラムを実行できる）状態になります。BOWの場合は、あらかじめWindowsが起動されている状態で、2.の部分をWindowsアプリケーションとしてロードします*4。その後で、動かしたいBSDプログラムのバイナリイメージをメモリ上に読み込み、制御をそのバイナリイメージに移

*4　すでにBOW上でほかのBSDプログラムが動いている場合は、BOWのロードは行われません。直接BSDプログラムの実行が始まります。

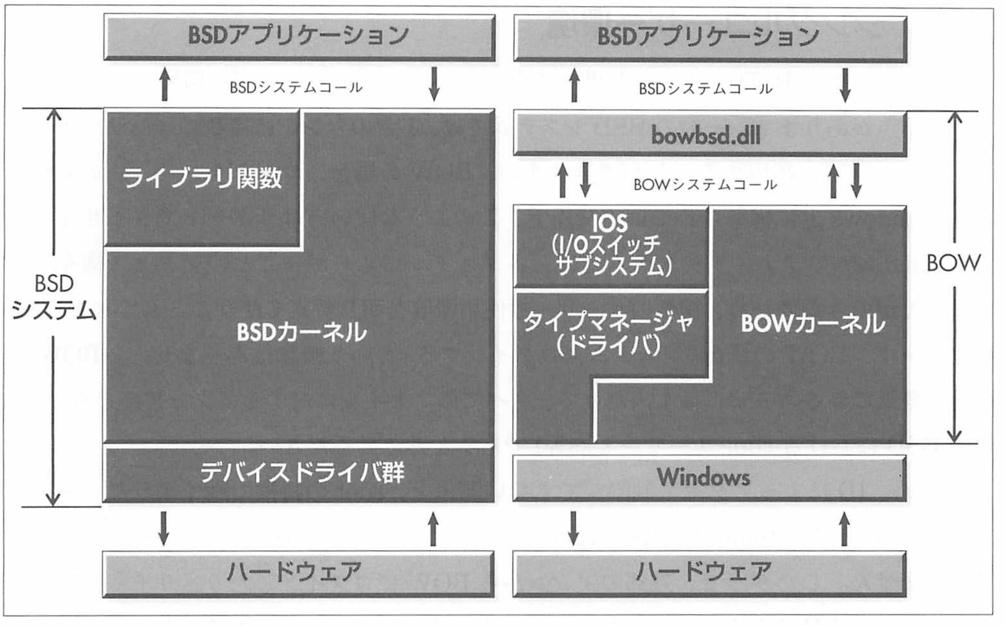


図 1-1　BOW と BSD システムの構成

します。BOW の場合、「BSD システムをブートする」という概念はありません。

前述したように、BOW は BSD のシステムコールを、バイナリレベルでエミュレートするシステムです。したがって、BSD プログラムから見ると、BOW は一般の BSD システムとまったく同じように映ります。たとえば、BSD ではファイルやデバイスに対する入出力を、read (2)、write (2) のシステムコールで行ないます。BSD プログラムがこれらのシステムコールを発行すると、BOW はこれを Windows の一連の入出力 API を使って処理し、その結果を BSD プログラムに返します。

複数の BSD プロセスを同時に実行したり、これらのプロセス間でデータのやりとりを行なうマルチプロセスの機能も、通常の BSD システムと同様に使えます。ただし、BOW は 16 ビットの Windows API (Win16 API) を使っているため、非同期的に複数のプロセスを切り替えることはできず、プロセスの切り替えは、BSD システムコールが発行されたタイミングに限られます。したがって、長時間システムコールを発行しないプログラムを実行した場合、問題が生じることがあります。

1.2.2 シングルユーザー環境

BOW が一般の BSD システムと異なる点の一つに、マルチユーザー環境をサポートしていない点があります。一般の BSD システムでは、1 台のマシンに複数のユーザーがログインして、同時に利用することができますが、BOW の場合、シングルユーザーの OS である Windows 上に構築されている関係上、このような使い方はサポートできません。また、Windows のファイルシステムでは、各ファイルにユーザーごとのアクセス権を設定することができないため、複数のユーザーで使用環境を切り替えて使うこともできません。したがって、BOW の場合「システムにログインする」という概念はありません。BOW の場合、登録できるユーザーは 1 人だけで、すべてのファイルに対するアクセス権をもつ(一般の BSD における root ユーザーと同等)であると考えてください。

ユーザー ID によるアクセス制限ができない関係上、root の特権で動くようなデーモンプログラム(とくに telnetd、ftpd、rpcd などのネットワーク関連デーモン)を動かすことはできません。したがって、外部のマシンから BOW にログインしたり、リモート実行するような使い方はできません。また、rpc、rsh、rlogin などの、いわゆる「r 系コマンド」は、ネットワーク全体のセキュリティ上の問題が発生するため、動作しないようになっています。ユーザー権限で動作するかな漢字変換サーバ(cannaserver、skkserv など)を動かすことは可能です。ただし、ほかのマシンから、BOW 上で動いている変換サーバにアクセスすることは原則としてできません。

1.2.3 各種デバイスのサポートについて

BOW は Windows の提供する機能を使って BSD 環境をエミュレートしている都合上、ファイルシステムや各種デバイスのサポートは、基本的に Windows に依存しています。したがって、次のような制限があります。

1. グラフィックデバイスを直接制御する X Window のようなインターフェイスは使えない。
2. ファイルシステムは Windows(DOS)のもの(FAT/VFAT ファイルシステム)を使っているため、ファイル名の付け方、ファイルの属性などは、FAT ファイルシステムの制限を受ける。Windows 3.1 で利用する場合、ファイル名の大文字・小文字の区別はなく(すべて小文字とみなす)、11 文字を超えるファイル名は切り捨てられる。ファイル名に関する規則については、168 ページを参照。ただし、シンボリックリンクやハードリンクのエミュレーションはサポートしている。また、

スーパーアスキー

Millenium より

http://www.ascii.co.jp/superascii/

ISO9660 Rockridge Extention形式のCD-ROMについては、BOW上のBSDプログラムからアクセスする場合に限って、BSDのファイルシステムと同じように使うことが可能。

3. 一般のBSD用に作成したローカルのファイルシステムをBOWからマウントして使うことはできない。ただし、PC NFSのようなネットワークファイルシステムをWindows側でマウントしている場合は、BOWからこのドライブにアクセスすることは可能。
4. 各種テープドライブをBOWから使用することはできない。
5. COM1、COM2などのシリアルポートを直接使うことはできない。したがって、モデムなどを使う通信プログラムをBOW上で実行することはできない。
6. ドライブを任意のディレクトリにマウントして使うことはできない。たとえば、Dドライブ全体を「/usr/local」にマウントするようなことはできない(Dドライブの内容は、/dfs/dというパスでアクセスできる)。
7. ネットワークの利用には、Windows用のTCP/IPネットワークドライバ(Winsock 1.0以上のAPIに対応のもの)を別途インストールする必要がある(Windows 95の場合は不要)。また、ネットワークドライバの種類によっては、機能に制限があったり、BOWでは動作しない場合もある。
8. ファイルシステムの仕様に関連して、オープン中のファイルのロック(flock(2)システムコール)はできない。flockシステムコールを使っているBSDプログラムについては、一部機能が使えないか、flockを使わないように修正する必要がある。

1.2.4 テキストファイルとバイナリファイルの扱い

BSDをはじめとしておもなUNIXシステムと、MS-DOSをベースとしたWindowsのシステムでは、テキストファイルの形式が異なっています。おもな相違点は次のようになります。

- 行末を示すコードが、UNIX系OSではLFのみなのに対して、MS-DOS系のシステムではCR+LFの2バイトで表現される。
- ファイルの終わりを示すコードの違い。
- 文字コードの違い。

これらの中で、とくに問題になるのは行末コードの取り扱いです。BOW上でBSDのプログラムを使う場合、MS-DOSの形式で作成されたテキストファイルを処理しようとす

ると、行末コードの取り扱いがうまくいかず、誤動作をする場合があります。mule などの一部のプログラムでは、UNIX 形式と MS-DOS 形式を自動的に判別して、適切に処理できるものもありますが、ほとんどのフィルター系ツールやテキストエディタは、このような機能はもっていません。

BOW は、Windows アプリケーションと BSD アプリケーションを同時に使うことのメリットを意図して作られているため、テキストファイルの扱いについて、特別な仕組を設けています（図 1-2）。

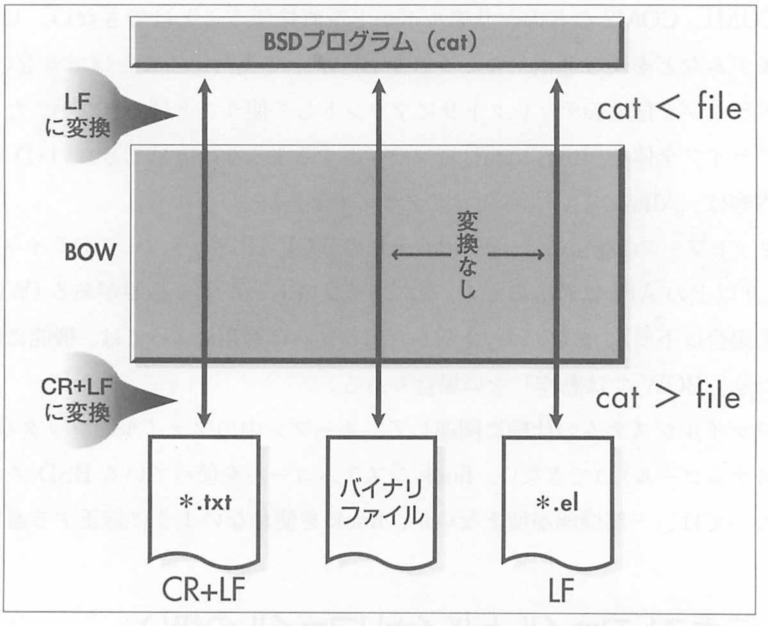


図 1-2　BOW のテキストファイル形式自動変換機構

この仕組により、特定の拡張子やファイル名をもったファイルへの入出力を行なう際に、ディスク上のファイル（MS-DOS 形式）は、自動的に UNIX 形式に変換されて BSD プログラムに渡されます。逆に、BSD プログラムからこのようなファイル名のファイルへデータを書き出すときには、UNIX 形式から MS-DOS 形式に変換されます。当然、これらのテキストファイルを Windows や MS-DOS 用のテキストエディタで編集しても問題ありません。

テキストファイルへの変換を行なうように指定していないファイルや、ファイル変換を行なうと不都合が生じるようなプログラム（tar など）については、ディスク中のイメージが、そのままプログラムに渡されます。テキストファイルの変換を行なうファイル名の指

定や、変換を禁止するプログラムの指定には、「BOW Configuration Tool」を使います。詳しくは、84ページを参照してください。

1.3 BOWの活用例

BSDシステムを初めて使う人は、BOWのどのように使えばよいのか戸惑うかもしれません。そこで、BOWで使えるBSDのプログラムや、BSDならでは(Windowsでは不可能な)使い方を簡単に紹介します。

1.3.1 BSDのアプリケーションを使う

BSDには、UNIXの黎明期から数多くのプログラマたちが作り上げてきた、膨大な種類のアプリケーションがあります。これらのなかでも、とくにテキストエディタは、UNIXのプログラマやユーザーたちが、自分たちがもっとも使いやすいと考えるものを作り、改良を続けてきました。

BOW開発の動機にもなった「emacs」は、テキストエディタとしての基本的な機能がしっかりしているだけでなく、「emacs lisp」というプログラミング言語によって、emacs上で動くアプリケーションを作ることもできる、拡張性にきわめて優れたエディタです。BOWでは、このemacsや、emacsを多言語拡張した「mule」を、一般のBSDシステムとまったく同じように使うことができます。たとえば、emacsのバッファ(ウィンドウのようなもの)の中でシェルを動かし、他のBSDコマンドを実行するといったマルチタスク機能も問題なく使えます。また、ネットワークなどで流通している各種emacs lispのプログラムも、ほとんどすべてがBOW上のemacsで使うことができます(図1-3)。

emacsと並んで多くのユーザーに使われている「vi」についても、vi互換のvimなどが動きます。vi互換のエディタは他にも複数存在しますが、これらのプログラムも、ソースファイルを入手してコンパイルすれば、BOW上で使うことができるでしょう。

BSDシステムには、テキストファイルを整形したり印字するツールもいくつか存在しますが、中でも「TeX」と「LaTeX」は、その機能の豊富さと印字結果の美しさで、UNIX上の文書整形アプリケーションの中で不動の地位を占めています。いわゆる「ワードプロセッサ」とは異なり、数多くのコマンドを覚えて、一種のプログラミングに似た作業をしなければならないので、初心者には使いづらいかもしれませんが、印字の結果は、そのまま印刷物の原稿として使えるほどの高い品質をもっており、論文誌や書籍の版下原稿作成に使われているほど本格的なものです(実際、本書を含め、アスキーの書籍の大半は、TeXを

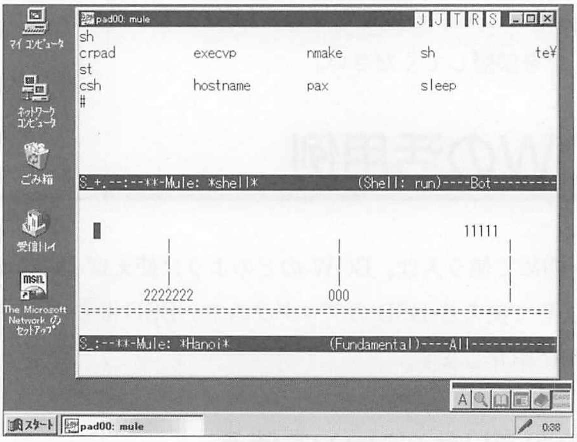


図1-3　BOWでemacsを使う

ベースとした組版システムで作られています)。

近年では、DOSやWindowsで使える$\TeX$システムも登場していて、機能的にもなんの問題もないようですが、BOWでは、最新のバージョン(pTeX 2.1)を使うことができます。また、Windows95上で使えば、ファイル名の長さによる制限もないため、一般のBSDとまったく同じように使えるでしょう。ただし、BOWはテキスト端末を使うアプリケーションしか実行できないので、$\TeX$の実行結果をディスプレイ画面上で見る(プレビューする)には、別途DVIファイルのプレビューアソフトを用意する必要があります。

1.3.2　フィルタ系ツールの活用

BSDとWindowsを使う上でのもっとも大きな違いは、コマンドライン・インターフェイスか、グラフィカル・インターフェイスかの違いです。BSDの場合、ユーザーは実行したいコマンド名をキーボードから入力します。Windowsでは、画面上のアイコンなどをマウスで指示してクリック(ダブルクリック)することで、プログラムを操作します。

画面上のアイコンやメッセージを見ながら操作できるグラフィカルインターフェイスは、直感的に操作できるのでとっつきやすいという利点があります。一方のコマンドラインインターフェイスでは、必要なコマンドの名前やコマンドラインの書式、オプションの名前や書式を覚える必要があるため、コンピュータの初心者には使いづらいとされています。しかし、コンピュータに行なわせたい作業の性質によっては、コマンドラインインターフェイスのほうが優れている場合もあります。たとえば、あるディレクトリ中にあるすべてのテキストファイルの中から、特定の文字列を含む行を抽出して、それをソートするといっ

た作業を行なう場合、GUI ベースのアプリケーションではかなり大変な作業になります。しかし、BSD のフィルタツールを組み合わせれば、次のコマンドラインを実行するだけですみます。

```
% grep "[文字列]" ./* | sort > result.txt
```

サブディレクトリの下に存在するファイルを含めて、同じような操作を行なうことも簡単にできます (csh を使っている場合)。

```
% find . -name "*" -exec grep "[文字列]" {}¥; | sort > result.txt
```

1.3.3 ネットワークアプリケーションの活用

BSD システムは、主にネットワークに接続されたミニコンやワークステーションで使われてきたため、さまざまなネットワークアプリケーションが使えます。特に、TCP/IP プロトコル関連のアプリケーションが充実しています。BOW は、Windows 用の TCP/IP ネットワークソフトウェア (Windows95 の場合は、Windows95 に付属の TCP/IP ドライバ) を併用することで、BSD 上で使われているネットワークアプリケーションの多くを使うことができます。ネットワーク上のワークステーションなどとの間でファイル転送を行なう「ftp」、他の UNIX ワークステーションなどにリモートログインする「telnet」などの基本的なものをはじめ、クライアント・サーバ型のかな漢字変換システム (wnn、canna など)、ネットワーク対応のゲームなどが動きます (ただし、X Window などのウィンドウシステムを必要とするものは動きません)。

BOW でのネットワークの利用については、155 ページで解説します。

1.4 BOWに必要な環境

ここでは、BOW をインストールして BSD アプリケーションを使うために必要になるハードウェア、ソフトウェアの構成について解説します。ただし、Windows が動く機種やソフトウェア環境は多岐に渡るため、お使いの機種やソフトウェアのバージョンによっては、ここで述べる構成でうまく動作しない場合も考えられます。ご自分の環境で BOW のインストールや起動ができなかった場合は、すでにインストールする他のソフトウェアの構成を変えたり、不要なデバイスドライバを外すなどしてから、あらためてインストールを試みてください。

1.4.1 ハードウェア環境

■ CPU

インテル製i386SX/DX、i486SX/DX、Pentiumあるいはこれらに相当するCPUで、Windows 3.1あるいはWindows 95が障害なく動くもの。i386SX/DX、i486SXの場合は、それぞれi387、i487（あるいは486ODP）など、数値演算コプロセッサ（FPU）機能の追加が必須（FPUがないと動作しません）。

■ メモリ

Windows 3.1の場合8Mバイト以上、Windows 95の場合12Mバイト以上必要。いずれの場合も、16Mバイト以上を推奨します。

Windows 3.1の場合は、Windows起動前の状態で、コンベンショナルメモリ（DOSアプリケーションが使う1Mバイト以下のメモリ領域）の空きが530Kバイト以上必要です。BOWのインストールや起動ができない場合は、不要なデバイスドライバの使用を中止したり、UMB領域へ移動する（config.sysで「device＝」となっているところを「devicehigh＝」に変える）などして、コンベンショナルメモリを空けてください。また、Windowsアプリケーションの中には、大量のコンベンショナルメモリを消費するものもありますので、BOWのインストールや起動を行なうときには、できるだけこのようなアプリケーションの併用は避けてください。

■ CD-ROM

Windows 3.1あるいはWindows 95で支障なく使えるCD-ROMドライブが必要です。Windows 3.1の場合は、MSCDEX 2.22以上からアクセスできるドライブ（MS-DOS 6.1以上、PC-DOS 6.1以上に付属するMSCDEXからアクセスできるものなら問題ありません）。Windows for Workgroup 3.11（英語版）やWindows 95の場合、ネットワーク経由でアクセスできる（「ネットワークドライブ」として割り当てられる）CD-ROMドライブからインストールを行なうこともできます。

1.4.2 ソフトウェア環境

MS-DOS 5.0以上とWindows 3.1、またはWindows 95が必要。Windows NTでは動作しません。Windows以外のOSのWindows互換モードでも動作する場合がありますが、動作の確認は一切していません。Windows 3.1では一部の機能が制限されます。また、

メモリ不足やネットワーク関連のトラブルも起こりやすいので、BOW 1.5はWindows 95での利用を強くお勧めします。

1.4.3　ネットワーク環境

BOW上でネットワークアプリケーションを使う場合は、以下のソフトウェアとハードウェアが必要になります。ネットワークアプリケーションを使わない場合には、必要ありません。ネットワーク環境の設定については、155ページを参照してください。

■ LAN環境で使う場合

Windows 3.1の場合は、Windows 3.1用のTCP/IPネットワークソフトウェア、および、そのソフトウェアでサポートされているネットワークカードが必要になります。ネットワークソフトウェアはWindows Sockets 1.1（Winsock 1.1）以上のAPIに準拠していることが必要です。個々のネットワークソフトウェアのWinsockへの対応状況については、それぞれのネットワーク製品の発売元へお問い合わせください。

Windows 95の場合は、Windows 95に付属のTCP/IPドライバが使用できます。ネットワークカードについては、Windows 95に対応しているものが必要です。

■ 電話回線経由のダイヤルアップIP接続で使う場合

Windows 3.1の場合は、ダイヤルアップIP接続が可能なTCP/IPネットワークソフトウェアとモデムが必要。Windows 95の場合は、モデムのみ必要です。

1.5　BOW 1.0からの追加機能

BOW 1.0からBOW 1.5で追加された機能について、簡単に解説します。見出しの後に「Win95」と書かれている項目については、Windows 95のみでサポートされる機能、「WFW」と書かれている項目はWindows for Workgroupでサポートされる機能です。

1.5.1　Long File Nameへの対応（Win95）

Windows 95では、従来のDOS、Windowsでのファイル名の制限（最大8文字+3文字の拡張子）が大幅に緩和されました（最大1バイト文字で255文字まで。ただし、パス名を含めて259文字まで）。Windows 95上でBOW 1.5を使う場合も、長いファイル名を使うことができます。ただし、Windows 95では、ファイル名に大文字と小文字を使うこと

ができますが、ファイルアクセスの際には、大文字と小文字は区別されません。したがって、同じディレクトリ中に、同じ文字列で大文字と小文字の違いしかないファイルを作ることはできません(「Makefile」と「makefile」、「README」と「Readme」は同じファイル名とみなされるので、同一ディレクトリ中には存在できない)。また、Windows 95 では日本語文字列(1バイトカタカナや全角文字)を含んだファイル名をつけることができますが、BOW からはこのようなファイルには基本的にアクセスできません。ファイル名の制限については、165 ページも参照してください。

1.5.2　シンボリックリンクとハードリンク

BOW 1.5 では、BSD のファイルシステムにおけるシンボリックリンク(ソフトリンク)とハードリンクのエミュレーションができます。したがって、BSD の ln コマンドや link システムコールを使うことができます。BOW のファイルリンクは、Windows のレジストリ機能を使って実現していますが、このリンクは BOW システムの中でだけ有効です。BOW の中で作成したリンクファイルを、Windows 側から削除すると、障害が発生する場合がありますので、注意してください。

1.5.3　ネットワークドライブへのアクセス(WFW、Win95)

Windows for Workgroup や Windows 95 からアクセスできる各種のネットワークドライブは、BOW からも使うことができます。ファイルマネージャやエクスプローラでネットワークドライブにドライブ名を割り当てておけば、BOW からは"/dfs/[割り当てたドライブ名]"というパスでアクセスすることができます(これは BOW 1.0 でも同様)。BOW 1.5 では、BOW Configure ユーティリティでホストを登録すれば、"/dfs/[ホスト名]"の形式でアクセスすることもできます。ただし、NetBEUI プロトコル以外を使った外部ドライブ(NetWare や PC/NFS など)については、ドライブ名を割り当てる方法のみサポートしています。

1.5.4　プロセス情報の取得

BOWPS ユーティリティで、プロセス情報の詳細を表示させることができます (図 1-4)。

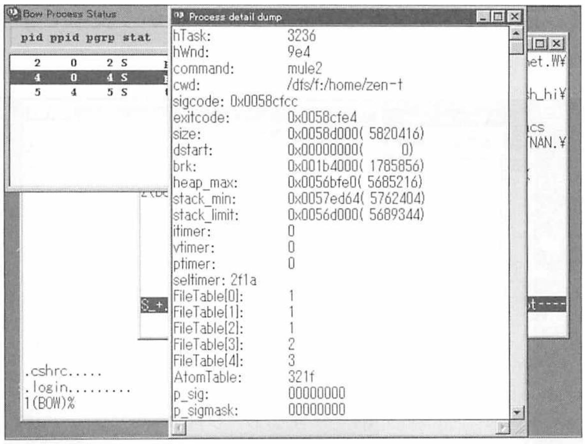


図 1-4　BOWPS でプロセス情報の詳細を表示する。

1.5.5　表示色のカスタマイズ

BOW Configure ユーティリティを使って、BOWPAD の表示色 (文字、背景、反転表示、強調表示など) を自由に変更することができます (図 1-5)。

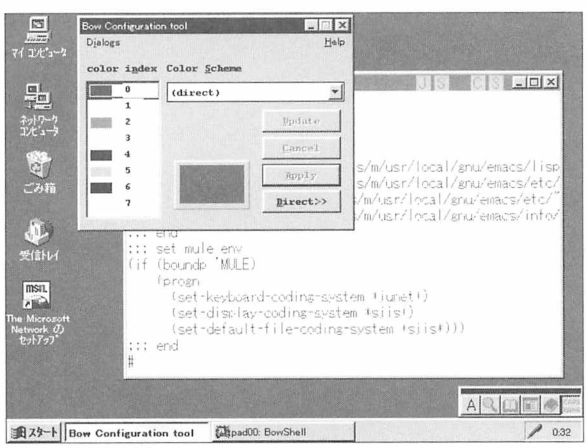


図 1-5　BOWPAD の表示色を変更する。

1.5.6 デバイスの追加サポート

専用デバイスファイルとして、wav 形式の音声ファイルを再生する"/dev/wave"、デバッグ用ウィンドウへ文字列を出力する"/dev/console"(図 1-6)が追加されました。/dev/console の出力ウィンドウは、BOWPS のコントロールメニューから表示できます。

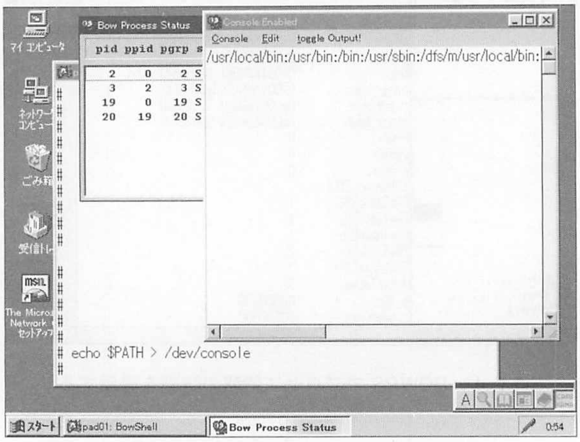


図 1-6 /dev/console への出力。

2章

セットアップ

BOWのセットアップ手順は、2つの大きな部分に分けることができます。まず、フロッピーディスクからのBOW本体のインストールを行ない、次に、インストールされたBOWシステムそのものを使って、CD-ROMに収められたBSDバイナリをインストールします。実際には、セットアッププログラムがこの2つの手順を自動的に進めます。画面上に表示される指示に従いながら、セットアップを進めてください。本章では、セットアップ前の準備、セットアップの手順、BSDプログラムを使うための環境設定、セットアップ時のトラブル対処法などを解説します。実際にセットアップを開始する前に、本章を一通り読んで内容を理解してください。

2.1 本章の前提事項

本章の説明では、BOWをインストールするハードウェアとソフトウェアが、**表2-1**のような構成になっていると仮定して説明します。これ以外の構成にインストールする場合はユーザーの構成に合わせてドライブ名やディレクトリ名などを読み変えてください。

表2-1 インストール構成

項　目	内　容
フロッピーディスクドライブ	A:
CD-ROMドライブ	M:
BOWインストール元	A:¥
BOWインストール先	C:¥BOW
BOWルートドライブ	C:
ユーザー名	bow
ホームディレクトリ	/root
ログインシェル	/bin/sh
Windowsディレクトリ	C:¥WINDOWS

BOWのルートドライブ(BSDプログラムをインストールするドライブ)は、何もファイルのない空のパーティションを割り当てることをお勧めします。これによりMS-DOSやWindowsのアプリケーションやデータなどと混在を避けることが容易になります。なお、MS-DOSやWindows 95の「SUBSTコマンド」を用いて、あるディレクトリから下のすべてのファイルとディレクトリを、独立したドライブに見せて使用することも可能ですが、動作の保証はいたしませんのでご注意ください。SUBSTコマンドについては、MS-DOSやWindowsのマニュアルを参照してください。

2.2 セットアップの準備

BOWは、通常のWindowsアプリケーションと同様に、ローカルのドライブ上に必要なファイルをインストールします。一般のBSDシステムのように、特別なファイルシステムを用意する(ハードディスクのフォーマットを行なう)必要はありませんが、BOWのシステムとWindowsのシステムでファイルシステムを共有するため、いくつかの注意すべき点があります。また、セットアップにはBOWそのものを動かして行なう処理も含ま

れているため、セットアップの実行時には、BOWが動く環境が整っていなければなりません。まず、BOWのセットアップを行なうにあたっての注意事項と、準備すべき事柄について解説します。

2.2.1　セットアップ上の注意事項

Windows 3.1にBOWをインストールする場合、特にメモリ環境について注意する必要があります。1章でも述べたように、BOWの動作には十分なコンベンショナルメモリの確保が必要になります。不要なデバイスドライバを削除したり、UMB領域に置くなどしてください。DOSに付属のメモリ管理ユーティリティを利用するのもよいでしょう。また、セットアップ作業を行なう際には、他のアプリケーションを起動しないでおいてください。

Windows 95の場合は、Windows 3.1に比べるとメモリに関する制限は緩和されています。しかし、スワップ領域が不足している場合などは、メモリ不足のためにセットアップが中断されてしまう場合もありますので、できるだけ他のアプリケーションを終了させた状態でセットアップを実行してください。

ハードディスクの容量不足にも注意してください。BOWのセットアップでは、すべてのBSDバイナリをハードディスクにインストールするか、CD-ROM上のバイナリファイルを利用して、ハードディスク容量を節約するかを選択できます。セットアップの途中でハードディスクの容量が不足しても、セットアップは中断されず、一見正常に終了してしまいます。セットアップ中は画面上に表示されるメッセージに注意して、「disk full」などのメッセージが表示されたら、空きディスクの容量やセットアップ先に指定したドライブ名を確認して、再度セットアップを行なってください。

2.2.2　インストールするドライブの決定

BOWのインストールに先立って、まず行なわなければならないのが、各種ファイルをインストールするドライブの決定です。

BOWのシステムは、「BOW本体」(Windowsアプリケーション)と、「BSDプログラムのバイナリファイルや設定ファイルとドキュメントファイルなどの集まり」の2つの大きなグループに分けられます。BOW本体は、「BOW」という名前のディレクトリにインストールされます。このディレクトリは、任意のドライブに割り当てることができます。

一方のBSDファイル群については、一般のBSDシステムと同様に、/etc、/bin、/usr、/dev、/varなどの複数のディレクトリに分かれてインストールされます。これらのディレクトリは、すべて同じドライブ上に作成されます。BOWでは、BSDバイナリなどが

インストールされるドライブのことを「ルートドライブ」と呼びます。ルートドライブは、DOS/Windowsで認識される任意のドライブに設定できます。ルートドライブに設定したドライブで、すでにBSDシステムで使われているものと同じディレクトリ名を使っている場合、そのディレクトリにBSDのファイルが上書きされます。したがって、BOWのルートドライブは、できるだけDOSやWindowsで使っているドライブとは別の、専用のドライブ(パーティション)に設定することをお勧めします。「BOW」ディレクトリを作成するドライブと、ルートドライブを同じドライブにすることも可能です(図2-1)。

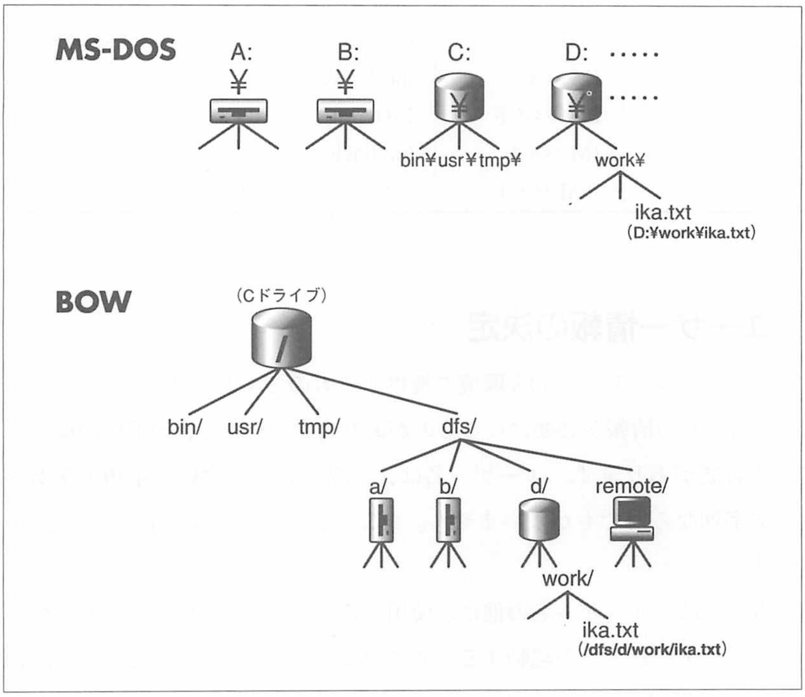


図2-1　BOWのドライブ構成

BOWやBSDでは、ユーザーの作業用のディレクトリ(ホームディレクトリ)を設定する必要があります。ホームディレクトリは、任意のディレクトリを指定することができるので、すでにDOS/Windowsで作業用のディレクトリを決めている場合には、そこをBOWのホームディレクトリとして設定することも可能です。BOWのセットアップを行なうと、ホームディレクトリに指定したディレクトリに、各種の初期設定ファイルが作られます。

BSDのバイナリファイルや各種設定ファイルなどは、すべてルートドライブにコピーされますが、ハードディスクの容量が足りない場合には、CD-ROM上のバイナリファイル

を実行するような設定にすることも可能です。表2-2に、各セットアップ方法で必要になるハードディスク容量をまとめておきます。

表2-2 インストールに必要なハードディスク容量

バイナリアーカイブ名	必要容量概算	説　明
basebin.tgz	26Mバイト	HDD上でBSDプログラムを実行するのに最低限必要な構成
minimum.tgz	5Mバイト	CD-ROM上のBSDプログラムのみを実行するのに必要な構成
mulebin.tgz	2Mバイト	muleの実行ファイル
mulelib.tgz	10Mバイト	muleのライブラリファイル(elispファイルなど)
jperl.tgz	1Mバイト	jperl一式
mhbin.tgz	10Mバイト	MH一式
jnh1013.tgz	1Mバイト	jnethack一式
ptex2.1.tgz	10Mバイト	アスキー日本語pTeX 2.1一式

2.2.3 ユーザー情報の決定

BOWはシングルユーザーのOS環境を提供するものですが、BSDプログラムの一部には、ユーザー名などの情報が必要になるものがあります。そこで、BOWの場合もユーザー名を決定する必要があります。ユーザー名は、一部の記号を除いた半角の英数字による8文字以内の文字列なら何でもかまいません。ただし、2バイトコード(全角文字)やカタカナは使えません。

ユーザー情報には、ユーザー名の他に、使用するシェル(ユーザーが入力したコマンドを解釈して、アプリケーションを起動するプログラム)の種類も含まれています。BOWではBSDシステムでよく使われている2種類のシェル(bash、csh)を選択できるようになっていますが、ユーザー情報に登録されたもの以外のシェル(tcsh、zshなど)を使うこともできます。

2.2.4　ネットワーク上のCD-ROMからインストールを行なう場合

LANで接続されたWindows 95マシンにBOWをインストールする場合、ネットワークで接続された他のマシンのCD-ROMドライブからBSDプログラムをインストールすることが可能です。この場合、あらかじめそのCD-ROMドライブにアクセスできるように準備しておく必要があります。Windows 95のLAN接続については、Windowsのマニュアルを参照するか、LANの管理者に問い合わせてください。ここでは、すでにLANによるネットワーク接続ができていることを前提とします。

■ 手順1

まず、「BSDプログラムCD-ROM」を、CD-ROMドライブを備えたマシン(ここでは「cdhost」というマシン名であると仮定します)に挿入します。マシンcdhost上で、CD-ROMドライブをネットワーク上で共有できるような設定になっていない場合([マイコンピュータ]のウィンドウで、CD-ROMドライブのアイコンが図2-2のようになっている場合)には、以下の手順で共有できるように設定します。アイコンが図2-3のようになっている場合は、すでに共有の設定がなされていますので、「手順2」に進んでください。

[マイコンピュータ]のウィンドウで、CD-ROMドライブのアイコンをマウスで選択し、[ファイル(F)]メニューの[共有(H)]を選択します。すると、図2-4のようなダイアログボックスが表示されるので、[共有する(S)]のボタンを選択し、[共有名(N)]のテキストボックスに適当な名前(ここでは「CDROM」)を入力します。[コメント(C)]のテキスト



図2-2　共有されていないドライブのアイコン

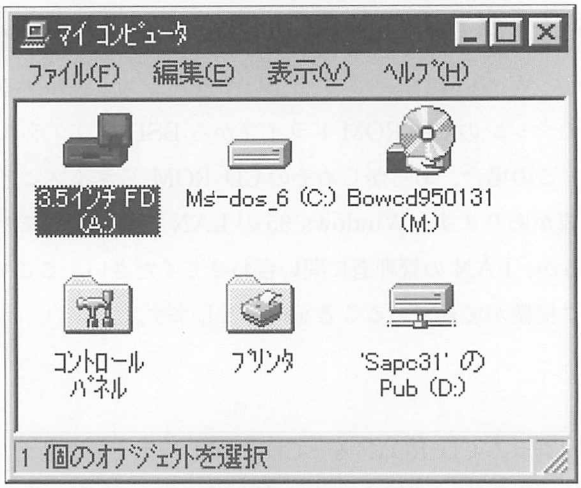


図 2-3　すでに共有の設定がされているドライブのアイコン

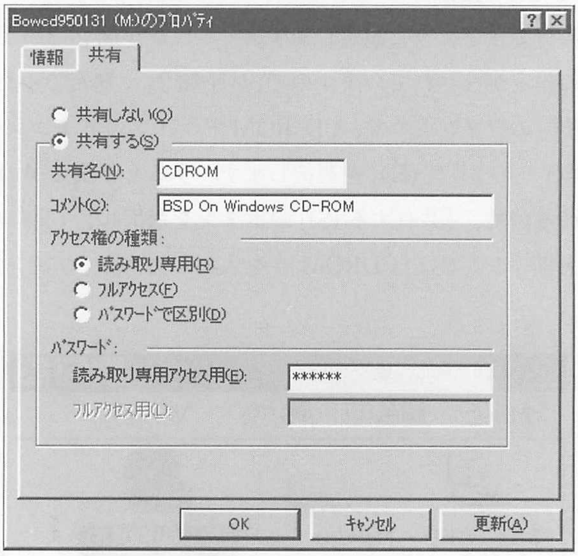


図 2-4　ドライブの共有を設定するダイアログボックス

ボックスは入力しなくてもかまいません。[アクセス権の種類] は [読み取り専用 (R)] を選択します。他人にアクセスされないようにしたい場合は、[パスワード] の [読み取り専用アクセス用 (E)] のテキストボックスに、適当なパスワードを入力しておきます。

■ 手順 2

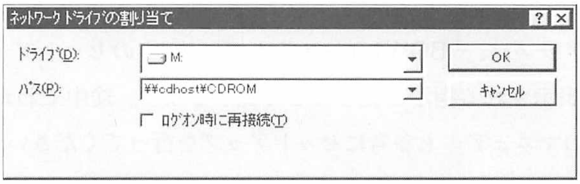


図 2-5　ネットワークドライブの割り当て

BOW をインストールしたいマシンで［エクスプローラ］のウィンドウを開きます。エクスプローラの［ツール(T)］メニューの［ネットワークドライブの割り当て(M)］を選択すると、図 2-5のようなダイアログボックスが表示されます。このダイアログボックスの［ドライブ(D)］では、CD-ROM を割り当てたいドライブ名（ここでは「M」ドライブと仮定します）を選択します。次の［パス(P)］の部分には、BSD プログラム CD-ROM を入れたマシンと共有名（手順 1 で指定したものと同じもの）を、次のような形式で指定します。

```
¥¥[マシン名]¥[共有名]
```

この例では、［マシン名］は［cdhost］、［共有名］は［CDROM］なので、［パス(P)］のテキストボックスには、

```
¥¥cdhost¥CDROM
```

と入力します。この CD-ROM ドライブに常にアクセスしたい場合には、［ログオン時に再接続(T)］のチェックマークを選択しておきます。BOW のインストールだけを行ないたい場合には不要です。

以上の設定で、リモートホスト上の CD-ROM ドライブからインストールを行なうことが可能になります。

2.3　セットアップの手順

BOW のセットアップは、「BOW1.5 システムディスク」のセットアッププログラムを起動して、画面上に表示される指示に従うことで完了します。途中でわからなくなった場合などのために、このマニュアルも参考にセットアップを行ってください。途中でセットアッププログラムや Windows のエラーメッセージが表示された場合は、その指示に従ってください。BOW Setup を途中で終了しなければならなくなった場合はその原因を解消して、手順を最初からやりなおしてください。

BOW の本体は付属の「BOW1.5 システムディスク」に入っています。この 3.5 インチのフロッピーディスクは 1.44M バイトでフォーマットされています。作業の前に、必ず BOW システムディスクを MS-DOS の DISKCOPY.EXE や Windows のファイルマネージャ (WINFILE.EXE) あるいはエクスプローラ (Windows 95 の場合) を使ってバックアップしておいてください。

2.3.1　セットアッププログラムの実行

「BOW1.5 システムディスク」を A ドライブに、「BSD プログラム CD-ROM」を M ドライブに挿入します（ネットワーク上のマシンにある CD-ROM からインストールする場合は、そのマシンの CD-ROM ドライブに挿入）。次に、[スタートメニュー] の [ファイル名を指定して実行 (R)] を選択し、図 2-6のようなダイアログボックスが表示されたら、[名前 (O)] のテキストボックスに「A:\SETUP.EXE」と入力し、[OK] ボタンをクリックします。「BOW Setup へようこそ」というメッセージ (図 2-8) が表示されますので、確認のうえ、[Continue] ボタンをクリックします。

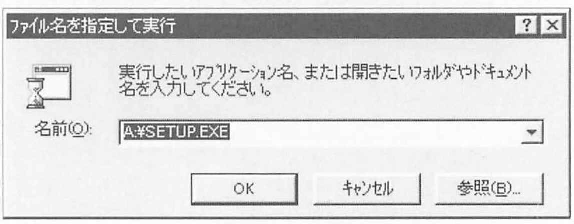


図 2-6　[ファイル名を指定して実行] ダイアログボックス (Windows 95)

Windows 3.1 の場合は、プログラムマネージャの［アイコン (F)］［ファイル名を指定して実行 (R)...］を選択し、ダイアログボックスが表示されたら、図 2-7のように［コマンドライン (C)］テキストボックスに「A:¥SETUP.EXE」と入力し、［OK］ボタンをクリックします。以下、Windows 95 の場合と同じ手順で進めてください。

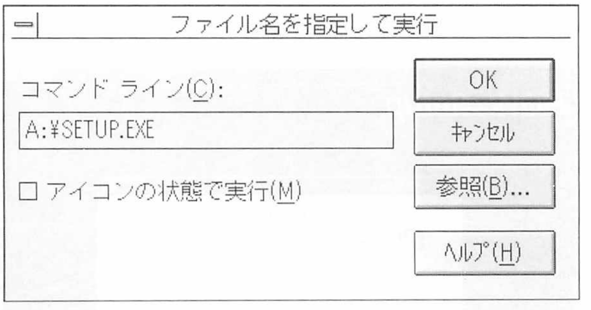


図 2-7　［ファイル名を指定して実行］ダイアログボックス（Windows 3.1）

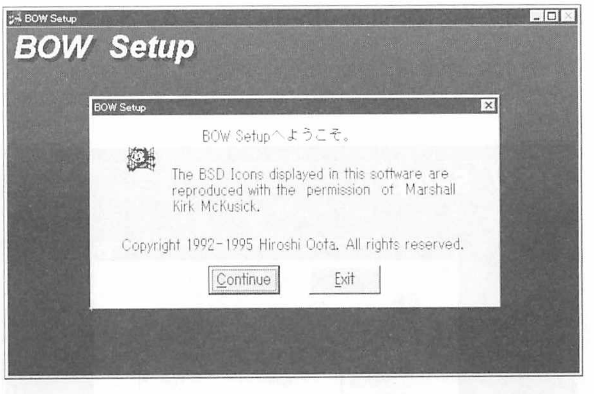


図 2-8　BOW セットアッププログラム起動画面

2.3.2　CD-ROMの確認

図2-9が表示されたら、[Continue]ボタンをクリックしてセットアップを続けます。もし、前の手順でBSDプログラムCD-ROMを挿入し忘れていると図2-10が表示されますので、CD-ROMを入れて[Retry]ボタンをクリックします。すると、図2-9が表示されるので[Continue]ボタンをクリックしてください。



図2-9　[Confirm CDROM]ダイアログボックス

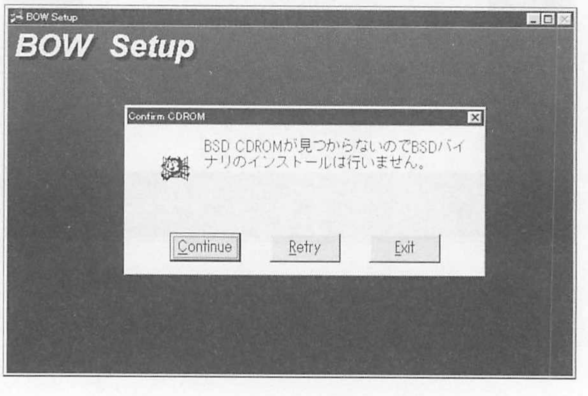


図2-10　BSDプログラムCD-ROMが入っていない場合の表示

2.3.3 BOWのセットアップ先の指定

[Set BOW directory]ダイアログボックス(図2-11)が表示されたら、BOWをセットアップするディレクトリをフルパス名で入力します。ディレクトリの初期値は「C:¥BOW」です。ここでは「C:¥BOW」と入力して[Continue]をクリックします。指定したディレクトリが存在しない場合は、自動的に作成されます。

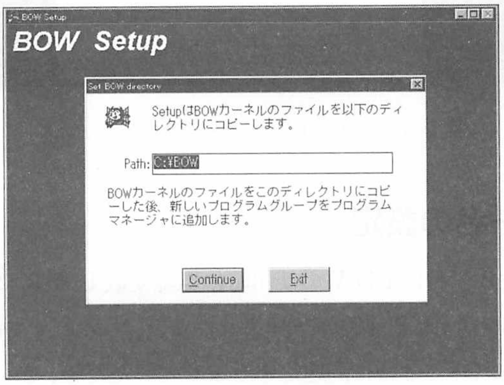


図2-11　[Set BOW directory]ダイアログボックス

BOWを組み込むディレクトリを入力したら、マウスポインタで[Continue]ボタンをクリックします。

2.3.4 ルートドライブの指定

ここで指定したドライブのルートディレクトリがBOWでのルートディレクトリ(/)になります。このドライブには、「2.4　BSDプログラムのインストール」でのインストール先になります。

[Root Drive]ダイアログボックス(図2-12)が表示されたら、[Select one]リストから「c:」を選択して[Continue]ボタンをクリックします。

このドライブとBOWをインストールするドライブは別になってもかまいません。ここで選ばれなかったドライブは、「/dfs/a、/dfs/b、…」などとして参照できます。Cドライブのルートに SYS というディレクトリがある場合は、「/sys」として参照されます。また、Dドライブのルートに DOS というディレクトリがある場合は、「/dfs/d/dos」という名前で参照されます。



図 2-12 [Root Drive] ダイアログボックス

2.3.5 ユーザー名の設定

ここで設定したユーザー名はBOWの使用時に環境変数USERに登録されます。[user name]ダイアログボックス(図2-13)が表示されたら、[Name]テキストボックスに「bow」と入力します。「bow」以外でも特に問題はありません。ただし、2バイトコード(漢字、記号)やカナは使用できません。入力が済んだら[Continue]ボタンをクリックします。

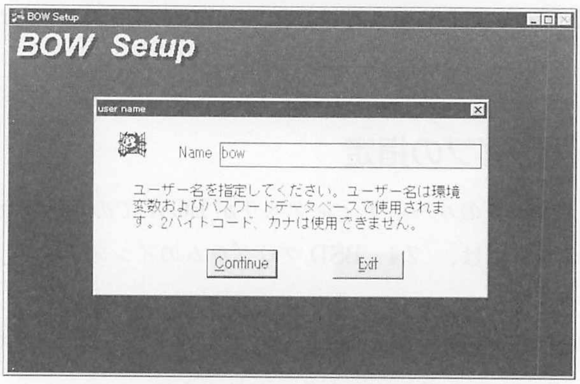


図 2-13 [user name] ダイアログボックス

2.3.6 ホームディレクトリの指定

BOWを起動したときのカレントディレクトリとなるホームディレクトリの指定を行ないます。このディレクトリにはシェルなどの環境設定ファイル(.login、.profile、.cshrc、.emacsなど)がコピーされます。

[home directory]ダイアログボックス(図 2-14)が表示されたら、[home directory]テキストボックスに「/root」と入力します。ホームディレクトリの初期値は「/root」ですがユーザーの好きなディレクトリに指定することも可能です(フロッピーディスクや MO ディスクなどのリムーバブルメディアには設定しないでください)。入力が済んだら[Continue]ボタンをクリックします。



図 2-14 [home directory]ダイアログボックス

2.3.7 ログインシェルの設定

ここで指定したシェルをプログラムマネージャに作成する[bow]グループに登録します。[login shell]ダイアログボックス(図 2-15)が表示されたら、/bin/sh または/bin/csh のどちらかを選択します。ここでは/bin/sh を選択して[Continue]ボタンをクリックします。



図 2-15 [login shell]ダイアログボックス

[login shell]ダイアログボックスでは、「/bin/sh」、「/bin/csh」のいずれかしか選択できません。これ以外のシェルを利用したい場合はインストールの終了後にユーザーが独自にアイコン登録するか、ここで選択したシェル上から利用するシェルを起動してください。付属のCD-ROMには、tcsh (/usr/local/bin/tcsh) とbash (/usr/local/gnu/bin/bash) が収録されています。

シェル

ユーザー入力したコマンドを解釈して実行するプログラム。MS-DOSのCOMMAND.COMに相当する。COMMAND.COMとの違いは、ワイルドカードの展開をシェルが行なう点である。コマンドラインの入力を容易にするために拡張されたものに、bash、tcsh、zshなどがある。

2.3.8 入力項目の確認

ここまでの手順で入力した項目を一覧表示する、[Confirm]ダイアログボックス(図2-16)が表示されます。入力した項目に間違いや変更が無い場合は[Continue]ボタンを、変更がある場合は[Back]ボタンをクリックします。[Back]ボタンをクリックした場合は、[Confirm CDROM]ダイアログボックスからの入力をやりなおすことができます。



図2-16 [Confirm]ダイアログボックス

2.3.9 ファイルのコピー

[DISK1]ダイアログボックス(図 2-17)が開きコピーの経過が表示されます。横に伸びるバーでセットアップの進行状況がわかります。ディレクトリやドライブの指定に間違いがあった場合は、[Exit]ボタンをクリックして、手順を最初からやりなおしてください。

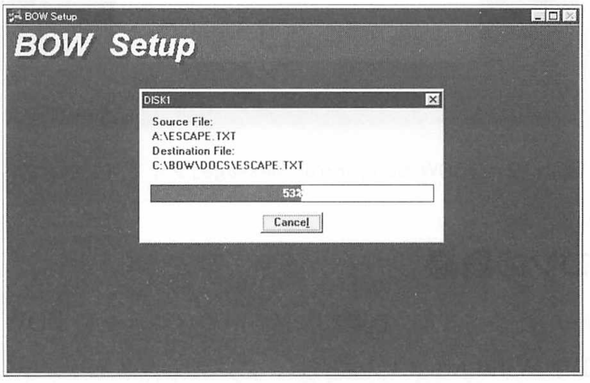


図 2-17　ファイルコピー経過画面

ディスクの容量不足が発生した場合には図 2-18のようなダイアログボックスが表示されます。図 2-18の例は「C:¥BOW¥JCONV.DLL」というファイルをコピーしているときに容量不足になったことを示しています。このような場合には[OK]ボタンをクリックし、[BOW Setup Error Message]ダイアログボックス(図 2-19)が表示されたら[OK]ボタンをクリックして、BOW のセットアップを終了します。セットアップをやりなおす際は不必要なファイルを削除し、ディスク容量を再確認したうえで行なってください。

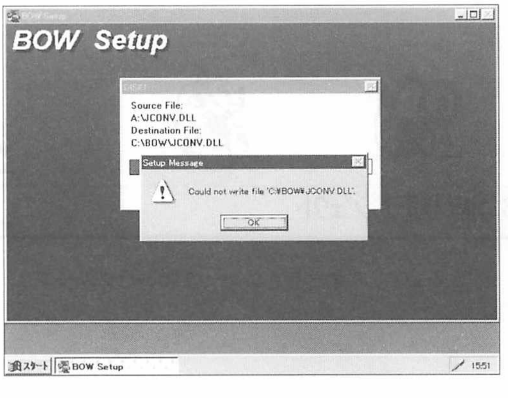


図 2-18　ファイルの書き込み不可能を示すダイアログボックス



図2-19　[BOW Setup Error Message] ダイアログボックス

2.3.10　アイコンの登録

ファイルのコピーが終了すると、自動的にスタートメニューにBOWのアイコンが登録されます(図2-20)。Windows 3.1の場合は、プログラムマネージャの[BOW]グループにアイコンが登録されます。



図2-20　スタートメニューの[bow]グループ

2.3.11 セットアップの終了

図 2-21のように「Bow の Setup は正常に終了しました。」というメッセージが表示されればセットアップは正常に終了したことになります。[OK] ボタンをクリックすると、引き続き、BSD プログラムのインストールが開始されます。



図 2-21　BOW セットアップ終了画面

2.4 BSDプログラムのインストール

BOW が動作する環境が整ったら、今度は BSD プログラムをインストールします*1。ここでは BSD のディレクトリ構成に沿ったインストールを行ないます。このインストールでは、/bin/sh や/bin/pwd のように決まったディレクトリにあることが期待されているプログラムが多いので BSD のディレクトリ構成に合わせたものとなっています。インストール後にディレクトリ構成を変更すると、一部のプログラムが動作しない場合があります。

2.4.1 パスワードデータベースの作成

BOW の環境では複数のユーザーを管理しませんが、/etc/passwd というファイルが正しく設定されていないと、うまく動作しないプログラム (mule など) がありますので、パスワードデータベースを作成します。

＊1　BSD プログラムのインストール時に「Can't load library.」などのエラーが出る場合は、コンベンショナルメモリが不足しています。26 ページを参照してください。



図2-22 [Create Passwd database]ダイアログボックス

[Create passwd database]ダイアログボックス(図2-22)が表示されたら、[OK]ボタンをクリックしてパスワードデータベースを作成します。

この作業を再度行ないたい場合は、プログラムマネージャの[bow]グループの[BSD setup]アイコンのコントロールメニューにある[Passwd DB](図2-23)で行なうことができます。

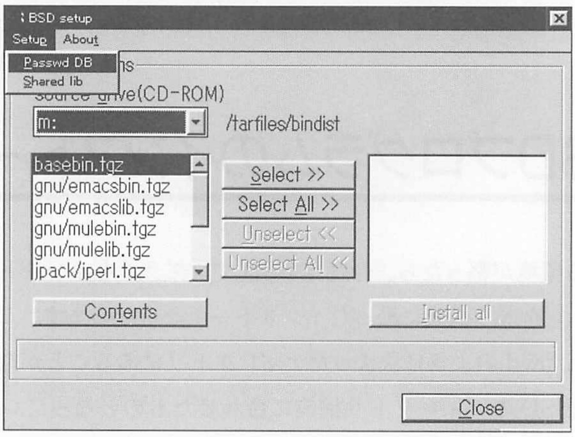


図2-23 [BSD setup]アイコンの[Passwd DB]

2.4.2 BSD setup

BSDプログラムは付属のBSDプログラムCD-ROMの/tarfiles/bindist以下に収録されているアーカイブ*2からインストールを行います。

---

*2 このアーカイブ(ファイル名の終わりが「.tgz」になっているもの)はtarコマンドで複数のファイルをひとつのファイルにまとめ、gzipコマンドで圧縮しています。

［BSD setup］ダイアログボックス（図 2-24）が表示されたら、左側のリストの中からインストールしたい BSD プログラムを選択します。インストールできる BSD プログラムの内容は表 2-3の通りです。

表 2-3　インストールできる BSD プログラム

ファイル名	内　容
basebin.tgz	4.4BSD-Lite 基本バイナリ
minimum.tgz	4.4BSD-Lite 最小バイナリ（CD-ROM 運用版）
gnu/mulebin.tgz	Mule 2.3（多言語拡張 emacs）
gnu/mulelib.tgz	Mule 2.3 ライブラリ（elisp、DOC、info ファイル）
jpack/jperl.tgz	jPerl 1.4
mail/mhbin.tgz	MH 6.8.3JP2c
misc/jnh1013.tgz	nethack-3.1.3-jnh-1.0.1.3
tex/ptex2_1.tgz	ASCII 版日本語 pTeX 2.1

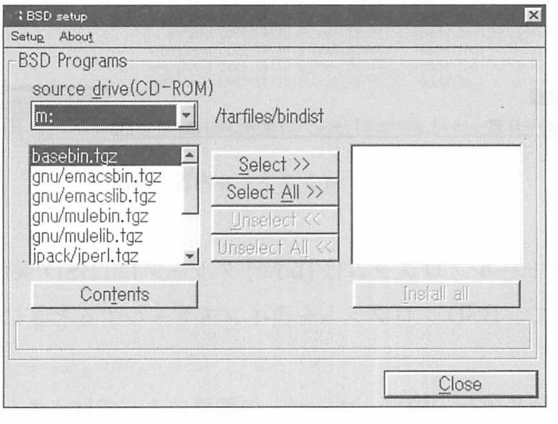


図 2-24　［BSD setup］ダイアログボックス

ここでインストールされるファイルの確認をしたい場合は「2.4.3　インストール内容の確認」へ、そのままインストールする場合は「2.4.4　アーカイブのインストール」へ手順を進めてください。

2.4.3 インストール内容の確認

インストールを実際に行う前にアーカイブの内容を確認をしたい場合は、確認するアーカイブファイルを選択して[Contents]ボタンをクリックします。BOW上で起動されているプロセスを表示するBOWPSとファイルのリストを表示する*3BOWPADが表示されます(図2-25)。リストの続きを表示させるにはリストを表示しているBOWPADをマウスカーソルでクリックし、スペースキーを押します。途中で表示を終了させる場合は q キーを押します。リストが最後まで表示し終わるとBOWPADは自動的に閉じます。

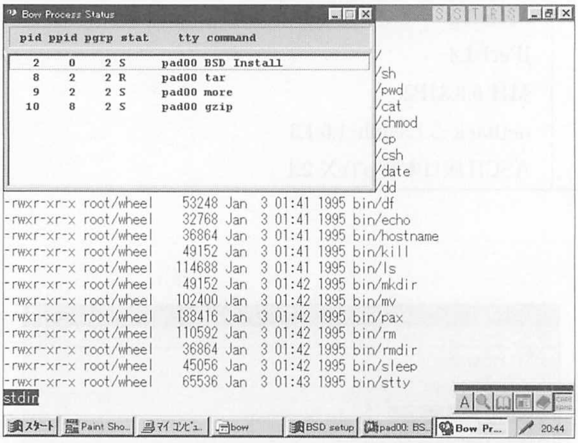


図2-25 インストール内容の確認

また、このインストールプログラムは[bow]グループに[BSD setup]アイコンとして登録されていますので、BSDプログラムを再インストールするときに、別個に実行することもできます。

インストールの確認や実行の際に、ユーザーの環境によっては、まもなくヒープサイズが足りなくなるという警告メッセージの「Heap size will exceed, Stop process as soon as possible. Current setting is 205K Bytes」が表示される可能性があります(〝205K〞はプロセスによって違う)。また、ヒープサイズが足りなくなった際の「Heap size exceeds. Aborting current process.」というメッセージと共に表示される[new value]に、次に同じプロセス名のプログラムが実行されたときのヒープサイズを指定することができます。[Change Setting]ボタンを押すと、変更した内容がBOW.INIに書き込まれます。変更したら、いったん[BSD setup]を終了し、再度実行します。

*3 実際には、tarコマンドのtオプションをCD-ROM上のusr/bin/moreで表示しています。

2.4.4　アーカイブのインストール

左のアーカイブリストから必要なアーカイブファイルを選択して右のリストに加えていきます。ここで重要なことは、CD-ROM を常時接続して、CD-ROM 上にあるコマンドを起動するか、CD-ROM から必要なファイルをすべてハードディスクにインストールするかを選ぶことです。前者なら「minimum.tgz」を、後者なら「basebin.tgz」をクリックして反転表示させます。CD-ROM から直接プログラムを利用すると、速度的には劣りますが、必要とされるハードディスク領域は最小限（5M バイト）で済みます。ハードディスクの残量に余裕がある場合（65M バイト以上）は、basebin.tgz を選択することをお勧めします。2 つのアーカイブのどちらかを必ず選択するようにしてください。

続いてその他のインストールをしたいアーカイブを選択します。Ctrl キーを押しながらクリックすることによって複数のアーカイブを選択できます。必要なファイルを選択し終えたら、[Select >>] ボタンをクリックして右のリストボックスに移動します（図 2-26）。

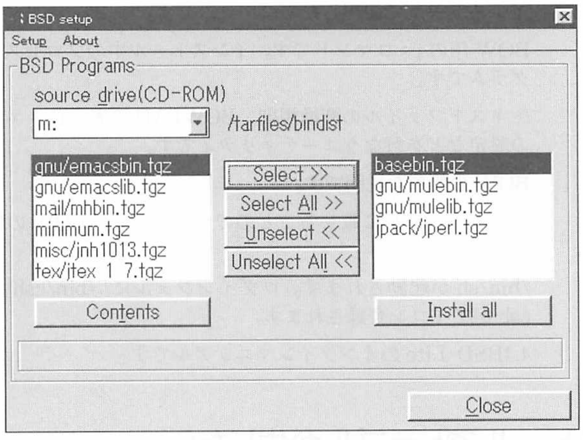


図 2-26　アーカイブの選択

[Install all] をクリックすると BOWPAD が起動されインストールが始まります。

インストールの処理の経過状況は、BOWPAD の中に表示されます。正常にインストールが行なわれている場合は、インストール中のファイル名のリストが連続的に表示されます。インストール先のディスク容量が不足した場合、「No space left on device」というメッセージが立て続けに表示されます。この場合は、ファイルのインストールは正常に行なわれていませんので、インストール先のディスク容量を確認したり、セットアップするファイルを減らすなどの対策が必要です。なお、このエラーが発生しても、「BOW のセッ

トアッププログラムは正常に終了した」というメッセージが表示されますので、注意してください。

2.4.5 インストールの終了

[Close]ボタンをクリックすと[BSD Setup]と[Bow Process Status](BOWPS)のダイアログボックスが終了し、インストールが完了します。

2.5 [bow]グループ

プログラムマネージャに作成された[bow]グループは次の通りです。各プログラムの詳しい内容については「3.2 BOW プログラムの使い方」を参照してください。

bow.ini	BOW の環境設定ファイルです。通常は直接編集することはありません。
BSD setup	インストールで使用したプログラムです。
ps	BOW 用の ps コマンドです。インストール中のプロセスを表示していたプログラムです。
BOW Configure	テキストファイルの変換規則、BOWPAD の表示色、ネットワークドライブの設定などを行なうユーティリティです。
BOW help	BOW の簡単な説明です。
release note	このマニュアルに載せることのできなかった、BOW の情報が書かれています。
sh	/bin/sh が起動されます。ログインシェルに「/bin/csh」を選択したときは、csh がアイコン登録されます。
4.4BSD-Lite	4.4BSD-Lite のオンラインマニュアルです。

2.5.1 インストールでトラブルが生じたら

インストールの途中でエラーメッセージが表示されたり、コンピュータがハングアップしてしまう、BSD のコマンドが実行できないなどの症状が出た場合は、メッセージの指示に従うか、次の各項目を検討して、再度インストールを試みてください。

■インストール時に他のアプリケーションが動いていないか

インストール作業中に、他のアプリケーションが動いていると、メモリが不足したり、ディスクへのアクセスが障害されたりする可能性があります。インストール作業中には、他のアプリケーションはすべて終了しておいてください。Windows のアプリケーションの

中には、画面にウィンドウやアイコンが表示されていなくても、バックグラウンドで動くものもあります。Windowsの起動時に自動的にアプリケーションを実行するような設定になっている場合に、BOWのインストールに障害が発生したら、一時的にアプリケーションの自動起動を中止して再度Windowsを起動し、BOWのインストールを再度行なってみてください。

■常駐型のデバイスドライバやプログラムを使っていないか

config.sysやautoexec.batで設定したTSR(常駐型)のプログラムがあったら、一時的に使用を中止してみてください。特に、Windows 3.1にBOWをインストールする場合、コンベンショナルメモリの空き容量がおおむね520Kbytesよりも少ないと、BOWを起動できない場合があります。

■BOW 1.0の環境に上書きインストールしていないか

Windows 95でBOW 1.5を使う場合、いままでWindows 3.1でBOW 1.0をインストールしてそのまま上書きインストールを行なった場合、次のような障害が起こる場合があります。

- BSDコマンドをコマンドラインから入力しても実行できない。
- BOW 1.0では小文字で表示されていたはずのファイルが、大文字で表示されるようになってしまった。
- BSDアプリケーションの動作が、以前と異なってしまう。

これらの症状の多くは、Windows 3.1とWindows 95では、ファイルシステムの仕様が変わっていることが原因になっています。Windows 3.1ではファイル名の長さや使える文字の種類に制限があったため、BOW 1.0では、BOW側からファイルにアクセスする場合に、ファイル名の変換を行なっていましたが、BOW 1.5をWindows 95で使う場合には、この変換の規則が変わっています。とくに、ファイル名の大文字小文字の変換については、BOW 1.5ではデフォルトでは行なわないようになっています。したがって、Windows 3.1で使っていた環境からWindows 95に移行した場合には、従来小文字の名前だったファイルが、大文字に変わってしまうため、BSDコマンドが実行できなかったり、動作が変わったりする場合があるのです。

Windows 3.1の環境を残したままWindows 95に移行している場合には、いったんBOW 1.0とBSDのファイルをすべて消去してから、改めてBOW 1.5とすべてのBSDファイルをインストールすることをお勧めします。このとき、ユーザー側で作成した各種

の設定ファイルやドキュメントファイル、ソースファイルなどをバックアップしておくことを忘れないでください。誤ってファイルを消去したことで生じる一切の損害に関して、当方では責任を負いません。

2.6 アンインストール

インストールしたあとでMS-DOSやWindowsのファイルとBOWやBSDプログラムのファイルを別々のパーティションに分ける必要が生じたり、BOWを再度インストールする場合のアンインストール手順を説明します。

2.6.1 [bow]グループの削除

Windows 95の場合は、[スタート]ボタンの[設定(S)][タスクバー(T)][[スタート]メニューの設定]の[削除(R)]ボタンを選択します。ここで表示されるメニューのツリー(図2-27)で[bow]という行をクリックして選択し、[削除]ボタンを押します。

Windows 3.1の場合、プログラムマネージャのbowグループを、[最小化]ボタンでアイコン化します。次にアイコン化された[bow]グループをクリックして選択します。最後にプログラムマネージャの[アイコン(F)][削除(D)]で削除します。

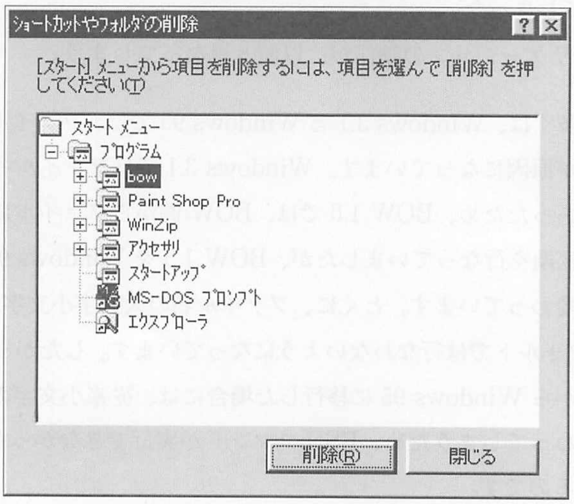


図2-27　スタートメニューから項目を削除

2.6.2 BOW.INIの削除

BOW の環境設定ファイルである、BOW.INI を削除します。BOW.INI は Windows ディレクトリに存在します。Windows 95 ならエクスプローラ、Windows 3.1 ならファイルマネージャ、DOS なら DEL コマンドなどで削除してください。この章のインストール例のパスは「C:¥WINDOWS¥BOW.INI」です。

2.6.3 BSDプログラムの削除

BSD プログラムの削除には注意が必要です。パーティションが同じでも、DOS や Windows のファイルと BSD のファイルが別々のディレクトリにインストールされている場合には、BSD のファイルだけを削除すればよいのですが、同じディレクトリに双方のファイルが混在していると、ファイルひとつひとつを確認していかなければなりません。BSD の実行形式ファイルは基本的に MS-DOS のファイル名のような拡張子(.BAT、.COM、.EXE など)は付きません。sh、sed、awk のようなスクリプトファイル(MS-DOS バッチファイルのようなもの)には、どのプログラムのスクリプトファイルかを区別できるように、「.sh」、「.sed」、「.awk」などの文字列*4をファイル名に付け加えている場合があります。削除するファイルの判断はユーザーの責任で慎重に行なうようにしてください。

---

*4 BSD のファイル名には「拡張子」という概念はありません。〝.〟は MS-DOS ではファイル名と拡張子を区切る役割であるのに対し、BSD ではファイル名の一部として利用できます(Windows95 の場合は、ファイル名の中でいちばん後方にある〝.〟に続く 1 文字以上の文字列が拡張子として扱われます)。そのため先頭に〝.〟を付けたり、複数の〝.〟を使ったりできます。詳しくは 165 ページを参照してください。

3章

初歩的な環境設定

BOW をインストールすると、選択したシェルや各種アプリケーションの基本的な設定（初期化ファイルや環境変数など）も自動的に行なわれます。初歩的な使用ではこのまま使い続けることもできますが、ここでは自分の好みにあわせて、使用環境を変更する方法について、簡単に解説します。

3.1　BOW側の基本的な設定 (Windows上で行なう操作)

BOW は Windows のアプリケーションの一種として動作します。このため、BOW を使う上での設定には、Windows 上で行なうものと、BSD プログラムに固有の設定を行なうものの 2 通りあります。まず、Windows 側で必要になる各種の設定について解説します。

3.1.1　アプリケーションアイコンの追加方法

BOW をインストールすると、タスクバーやプログラムマネージャに「BOW」という項目が作成されます。この中には、BOW のヘルプや [BOW configuration]、[BSD Setup]、そしてインストール時に選択したシェル ([sh] や [csh]) を起動するためのアイコンが作成されているはずです。これらのアイコンに加えて、ユーザーが新しいアイコンを追加して、BSD アプリケーションを直接起動できるようにすることが可能です。

アイコンの追加は、Windows 95 の場合はタスクバーの [設定 (S)] [タスクバー (T)] で、Windows 3.1 の場合は、プログラムマネージャの [アイコンの追加] メニューで行ないます。

Windows 95 の場合、次の手順で行ないます。

1. タスクバーの [設定 (S)] [タスクバー (T)] を選択する。
2. **図 3-1**のようなダイアログボックスが表示されたら、[スタートメニューの設定] のタブをクリックし、[追加 (A)] ボタンをクリックします (**図 3-2**)。次に表示されるダイアログボックスの [コマンドライン (C)] のテキストボックスに、次のような書式で BSD プログラムの起動コマンドを入力します。

```
C:¥BOW¥BOW.EXE [-n プロセス名] [-a] [-q] [-gXxY] フルパス名 [オプション...]
```

ここで、**フルパス名**には、BSD におけるコマンドのフルパス名を入れてください (DOS／Windows の形式ではありません)。**オプション**は、BSD コマンドに対してのオプションです。-n で指定する**プロセス名**は、ウィンドウのタイトルバーや BOWPS のリストに表示させたいプロセスの名前を指定します。-a オプションを指定すると、BSD コマンドが終了すると、同時にウィンドウも閉じます。-q を指定すると、ウィンドウを開かずにコマンドを実行します。また、-g オプションで実行時のウィンドウサイズを指定することもできます。

表示されるアイコンの変更については、Windows のマニュアルを参照してください。

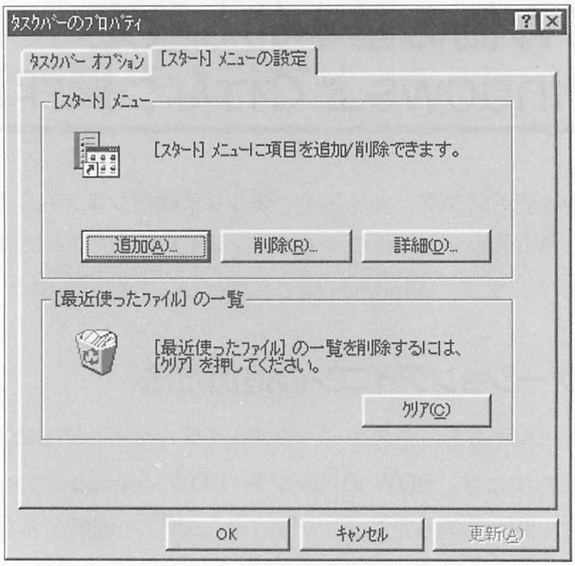


図 3-1　タスクバーの設定ダイアログボックス

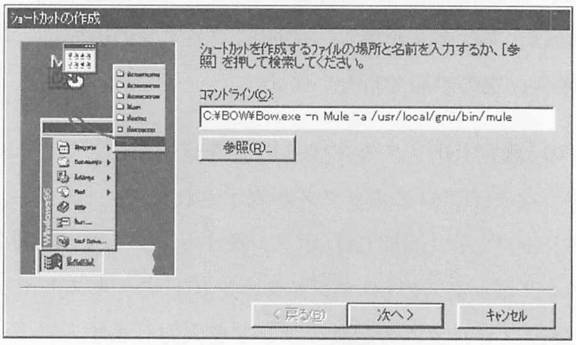


図 3-2　タスクバーへの項目の追加

3.1.2　漢字コードの設定

BOW では、BSD プログラムとのやり取りの際に使う漢字コードの種類を、ウィンドウごとに変更することができます。設定できるのは、キーボードから BSD のプロセスに渡されるデータと、BSD のプロセスから BOWPAD に渡されるデータの 2 つです。

手順を説明します。コントロールメニューから [Kanji Code...] を選択すると漢字コードを設定するためのダイアログボックスが表示されます。

選択できる漢字コードはシフト JIS、JIS (7bit JIS、Junet)、EUC の 3 種類です。[Keyboard Code]、[Display Code] に利用する BSD プログラムに合わせてコードの設定を

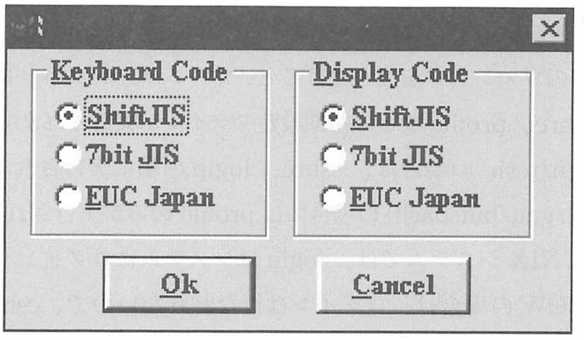


図 3-3　漢字コード変更ダイアログボックス

行ってください。ここで設定した内容は、プログラムマネージャあるいはタスクバーに登録されているアイコンの、BOW.EXE のコマンドラインに「-n」で指定されているプロセス名ごとに管理されています。このため、次回から同じ名前で起動された BOWPAD にも、同じ漢字コードが設定されるようになります。

3.1.3　BOW.INIについて

BOW のシステムと、BOW 上で動く BSD プログラムの動作を設定するのが、Windows のディレクトリ (C:\WINDOWS など) にある「BOW.INI」というファイルです。デフォルトの設定のまま BOW を使う場合は、このファイルの設定を変える必要はありませんが、高度な使い方をしたり、特別な BSD プログラムを動かしたい場合などに、このファイルの設定を変えなければならないことがあります。BOW.INI の設定の詳細ついては、91 ページを参照してください。

3.2　BSDプログラムの基本的な設定

BOW を起動して BSD プログラムが動きはじめると、その BSD プログラムは、本物の BSD システムで動いている場合とまったく同じ動きをするようになります。BSD プログラムの中には、環境変数や専用の環境設定ファイルなどを使って、その動作の方法を設定できるようになっているものがあります。BOW でこれらのプログラムを使う場合でも、この設定方法は一般の BSD システムの場合とほとんど同じです。ただし、環境変数への設定が必要なプログラムの場合、一般の BSD システムでは、.login、.cshrc、.profile などのファイルへ記述しますが、BOW の場合は、BOW.INI に記述することもできます。

3.2.1 シェル

BOW上でシェルのウィンドウを立ち上げると、ホームディレクトリに指定した場所に存在する、.login、.cshrc、.profileなどの初期設定ファイルが実行されます。csh(/bin/sh)やtcsh(/usr/local/bin/tcsh)の場合は、.cshrc、.loginの順に実行されます。sh(/bin/sh)やbash(/usr/local/gnu/bin/bash)の場合には.profileのみが実行されます。csh系のシェルの場合、一般のUNIXシステムでは、.loginはシステムにログインするときに、一度だけ実行されます。BOWの場合は、ログインは行なわれないので、cshのウィンドウが開かれるときに、常に.cshrcと.loginの両方が実行されます。あるアプリケーションの実行中に、子プロセスとしてcshを実行する場合(たとえば、emacsやmuleのshellモードなど)には、.cshrcのみ実行されます。

これらのシェル設定ファイルには、コマンドサーチパスなどの環境変数の設定、シェル変数の設定(csh系のみ)などを記述します。通常のUNIXシステムでは、ユーザーがログインすると、かならずシェルが起動し、そのシェルから各種アプリケーションを起動します。BOWの場合は、プログラムマネージャやスタートメニューにアイコンを登録することで、直接アプリケーションを起動することもできますが、この場合、.login、.cshrc、.profileなどに記述された設定は反映されません。このため、各種アプリケーションで必要になる環境変数の設定を、BOW.INIの[Environments]セクションに記述できるようになっています。たとえば、コマンドサーチパスに「/home/bin」を追加したい場合には、BOW.INIの[Environments]セクションの[PATH]の行を、

```
PATH=/home/bin:/usr/local/bin:/usr/bin:/bin
```

のように変更します。ここで設定した内容は、shやcshを実行する場合にも有効になります。BOWの場合通常は、すべてのアプリケーションに共通する設定内容は、BOW.INIに記述したほうがよいでしょう。

3.2.2 emacs / muleの設定

emacsやmuleが起動されると、まず、/usr/local/gnu/emacs/lisp/default.el(emacsの場合)、あるいは/usr/local/gnu/lib/mule2/site-lisp/default.el(muleの場合)が、次にホームディレクトリにある「.emacs」というファイルがそれぞれ実行されます。このファイルは、「emacs lisp」という言語で記述されています。BOWをインストールすると、これらのファイルには、あらかじめBOWで使う場合にもっとも適した設定が記述されています。各自でemacsやmuleをカスタマイズする場合には、ホームディレクトリの.emacs

ファイルの内容を編集してください。

emacs や mule を使う場合でもっとも問題になるのが漢字コードの設定です。BOW では、Alt キー(PC98 シリーズでは GRPH キー)を、emacs の Meta キーとして使うことができるようになっています。この機能を使うには、emacs や mule を起動する BOWPAD のキーボード漢字コード(keyboard code)の設定が「7bit JIS」になっている必要があります。この漢字コードの設定は、BOWPAD 側と emacs/mule 側の両方で行なう必要があります。BOW をインストールしたときに作成される.emacs には、この設定があらかじめ行なわれています。また、ウィンドウ名が「nemacs」や「mule2」となっている BOWPAD については、あらかじめ漢字コードの設定が前述の通りになっていますので、とくに変更する必要はありません。しかし、シェルのウィンドウから emacs/mule を起動する場合や、ウィンドウの名前を変えた BOWPAD で emacs/mule を起動する場合には、この設定が生かされませんので、その BOWPAD の漢字コードの設定を変える必要があります。なお、BOWPAD の漢字コードの設定を変えると、次に同じウィンドウ名の BOWPAD を開いたときに、変更が反映されます。シェルのウィンドウで漢字コードを変更した場合には、注意してください。

3.2.3 その他のアプリケーションについて

BSD のアプリケーションには、ユーザーの利用環境や利用目的に応じて、各種の設定を変更できるようになっているものがあります。これらの設定方法は、

- ユーザーのホームディレクトリに設定ファイルがあるもの。
- 各アプリケーションごとに指示されたディレクトリに設定ファイルがあるもの。
- 環境変数などで設定を変更できるもの。
- コンパイル時に設定を指定するもの。
- 上記の各種設定方法の組み合わせ。

など、さまざまな方法があります。BOW 上でこのようなアプリケーションを使う場合、設定の方法は、基本的に一般の BSD システムの場合と変わりません。各アプリケーションに付属するドキュメントや参考文献を参照して設定を行なってください。ただし、環境変数による設定が必要なものについては、本章の 64 ページでも述べたように、.cshrc、.login、.profile などによる設定ではなく、BOW.INI による設定を行なった方がよいでしょう。

3.3 ネットワークの設定について

BOWでは、Windows 95のネットワーク機能を使うことで、以下のようなことが可能になります。Windows 3.1では、別途Windows用あるいはMS-DOS用のネットワークソフトウェアを用意することでこれらの機能が使えるようになりますが、ネットワークソフトウェアの種類によっては一部の機能が使えない場合もあります。

■ TCP/IPプロトコルによる通信

telnet、ftp、メール、ニュースなどのネットワークアプリケーションの利用、DNSなどのネットワーク情報管理機構の利用。Windows 3.1では別途TCP/IPネットワークソフトウェアが必要。

■ Windowsがサポートするファイル共有機構の利用

Windows NTなどのファイルサーバーへのアクセス。Microsoftネットワークによる、ほかのマシン上のファイルへのアクセス。PC-NFSによるUNIXホスト上のファイルへのアクセス。ネットワーククライアント機能のないWindows 3.1では不可。

ネットワークの設定は、お使いのネットワーク環境の種類や構成によって変わってきます。詳しくは、本マニュアルの155ページを参照してください。

4章

BOW固有の機能

BOWの目的は、Windows上で一般のBSDシステムと同じようにBSDプログラムを使えるようにすることです。したがって、BSDプログラムそのものの使い方に関しては、一般のBSDシステムの場合とほとんど違いはありません。しかし、BOWでは、WindowsやMS-DOSとの相互運用が可能であるため、BOW固有の設定や操作があります。また、BOWの唯一のユーザーインターフェイスである「BOWPAD」では、各種の独自機能が使えるようになっています。本章では、BOW固有のこれらの機能とその使い方について解説します。BOWを初めてお使いになられる場合は、かならずお読みください。

4.1 WindowsおよびMS-DOSとの連携

BOWがWindowsやDOSと関わる部分は2つあります。ひとつはDOSのファイルシステムを使用したデータの共有です。ふたつめはBSDの環境からWindows、DOSプログラムを起動する"ランチャー"的な使い方です。このようにBOWはBSDのプログラムを動作させる以外にもWindows、DOSとの連携がうまく行なわれるように配慮されています。

表4-1に、BOWで固有の機能を実現させるために用意された、DOSコマンドとBSDコマンドの一覧を挙げます。本章を読む際の参考にしてください。

表4-1　BOW固有のコマンド

コマンド名	説　明
DOSWRAP.EXE	BSDプログラムとDOSプログラムの間で入出力をリダイレクトするDOSプログラム
doscnv	テキストファイルの行末を、UNIX形式とDOS形式で相互変換するBSDプログラム
BSD2DOS.EXE	BSD形式のファイル名をDOS形式のファイル名に変換するDOSプログラム
DOS2BSD.EXE	DOS形式のファイル名をBSD形式のファイル名に変換するDOSプログラム
execvp	BSDプログラムからDOSプログラムをアイコン化した状態で起動するBSDコマンド
crpad	新規のBOWPADを開き、標準入力の内容を表示するBSDコマンド
manbow	BSDのオンラインマニュアルをWindows 3.1のヘルプファイルに変換するDOSコマンド。別途HC31.EXEが必要(BOWには付属しない)

4.1.1 データの共有

BOWはDOS/Windowsのファイルシステムを使用しているので、ファイルを相互に利用することができます。BSDとDOSでは、テキストファイルの形式、とくに行末を表わすコードが異なっていますが、BOWでは正規表現を使ってファイル名(すなわちファイルの種類)を判別することにより、テキストモード、バイナリモードのどちらでファイルをオープンするかを区別できるようになっています。正規表現の指定には後述の[BOW Configuration]アイコンを使います。

また、BSD のデバイスに抽象化できるもの (read(2)、write(2)、ioctl(2) などのファイルオペレーションによって処理できるもの。キャラクタデバイスとして利用できるもの) は、デバイスドライバを作成すれば利用可能になります。デバイスドライバは DLL として BOW に組み込みます。BOW 用の DLL の作成方法については、本マニュアルでは触れません。

4.1.2 WindowsおよびMS-DOSプログラムの起動

BOW に実装されている BSD システムコール execve(2) は、Windows、DOS プログラムを実行できるように拡張されています。実際には Windows API の ShellExecute を呼び出しているので、.COM ファイル、.EXE ファイル、.PIF ファイルのほか、.WRI、.BMP などの、アプリケーションと関連づけられたドキュメントファイルを実行することもできます。

次に BSD プログラムから DOS プログラム、Windows プログラムを利用する方法をいくつか紹介します。

■ BSD プログラムから MS-DOS プログラムを利用する

BSD プログラムと DOS プログラムの間でデータをやり取りする場合には、doswrap.exe を使って標準入出力のリダイレクトを行ないます*1。

/bin/doswrap.exe は標準入出力、エラー出力をリダイレクトし、コマンドで指定された DOS プログラムを実行します。

使用法は次の通りです。

```
doswrap.exe [-n file|+n file] コマンド [args ...]
```

-n	n 番のファイルハンドルを file にリードモードでリダイレクトする。
+n	n 番のファイルハンドルを file にライトモードでリダイレクトする。
file	ファイル名の指定又は&に続く自然数でファイルハンドルを指定する。

ファイルハンドルを指定する場合は、すでにリダイレクトされているファイルハンドルを指定する必要があります。

$BOWDIR/samples/bin/nmake*2 は doswrap を使用し BSD プログラムから NMA

*1　このプログラムのソースコードは、bow/unsupported にインストールされています。

*2　BSD では「$環境変数」とすることで環境変数の内容を参照することができます。MS-DOS のバッチファイルで使われる「%環境変数%」にあたります。

KE.EXEを実行するシェルスクリプトです。たとえば、emacsから「M-x compile [Enter] nmake」*3を実行すると、コンパイル結果をemacsのバッファへ取り込めます。

このスクリプトではDOSプログラムの出力は非同期には実行されていません。したがってDOSプログラムがコンソールからのキー入力を要求すると、入力待ちのまま停止するので注意が必要です。

リンク時にファイルが見つからなかった場合に備えてmakefile中のLINK.EXEのオプションに/batchを指定することを強くお勧めします。

■ ExcelからBSDのシェルスクリプトを呼び出し、結果をクリップボード経由でExcelに貼りつける

この例では$BOWDIR/samples/excelの下にあるbowsynch.dll、bowsynch.xlmを使用します。bowsynch.dllはBSDプログラムと同期をとる機能をExcelに追加するDLLです。bowsynch.xlmはExcelのマクロファイルです。

次に具体的な手順について説明します。はじめに次の様なシェルスクリプトを/root/xltestとして作成してください。[TAB]は[TAB]キーを入力することを意味します。

```
#!/bin/sh
exec >/dev/clipbrd.txt 2>&1 < /etc/passwd
sed -e 's/:/ [TAB] /g' |doscnv -t
```

Excelの［ファイル(F)］［オープン(O)］で$BOWDIR/samples/excel/bowsynch.xlmをオープンします。［ツール(O)］［マクロ(M)...］［開始(R)］で、/root/xltestを実行するexec_bsdマクロを実行し、指示にしたがっていくとsheet1にパスワードファイルの内容が貼りつけられます。

/root/xltestの中で使っている、doscnvコマンドは行末の変換処理(CR/LF⟷LF)を行うBSD上のプログラムです。標準入力を処理して標準出力に書き込みます。変換方法は次に示すオプションで指定します。

doscnv -t　LFからCR/LFへ変換する
doscnv -f　CR/LFからLFへ変換する

＊3 本マニュアルでは、[Return]キー、[Enter]キーのいずれも、[Enter]キーとして表示しています。

4.2 BOWプログラムの使い方

ここではBOWの中心的なプログラムのBOWPADとBOWの運用を支援するツールの説明をします。

4.2.1 BOWPAD

BOW 環境を利用する上で一番よく使うのが BOWPAD(図 4-1) です。BSD コマンドの実行は基本的にこの BOWPAD 上に起動したシェルから行います。BOWPAD は BOW を使う上でユーザーとやりとりを行なう際の、唯一のインターフェイスになります。BOWPAD を使うことが BOW を利用することといってもよいでしょう。

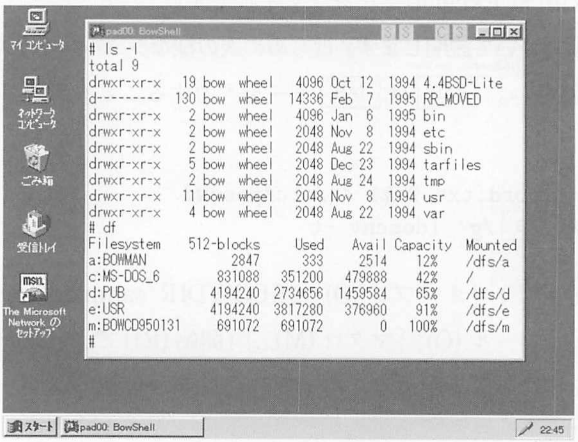


図 4-1　BOWPAD

BOW は BSD と同じくキャラクタベースのプログラムを対象にしています。BOWPAD は基本的には ANSI ターミナル互換の機能を提供しますが、Windows 向けの機能も付加されています。BOWPAD はユーザーが直接起動するのではなく、デバイスファイル(/dev/*) がオープンされるときに自動的に起動されます。

以降で説明する BOWPAD の機能はコントロールメニュー(図 4-2、図 4-3)または BOW PAD の上でマウスカーソルの右ボタンをクリックすると表示されるメニュー(図 4-4)から設定できます。

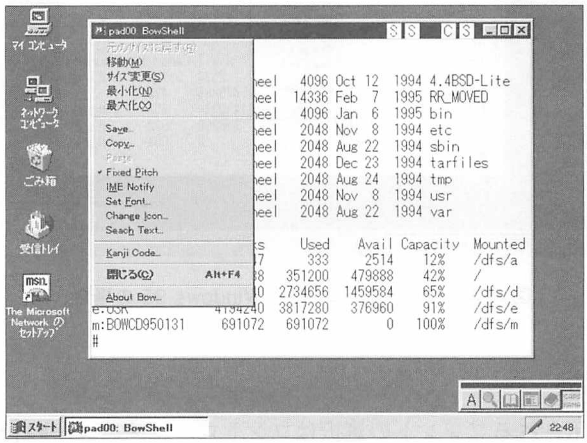


図 4-2　BOWPAD のコントロールメニュー(Windows 95 の場合)



図 4-3　BOWPAD のコントロールメニュー(Windows 3.1 の場合)

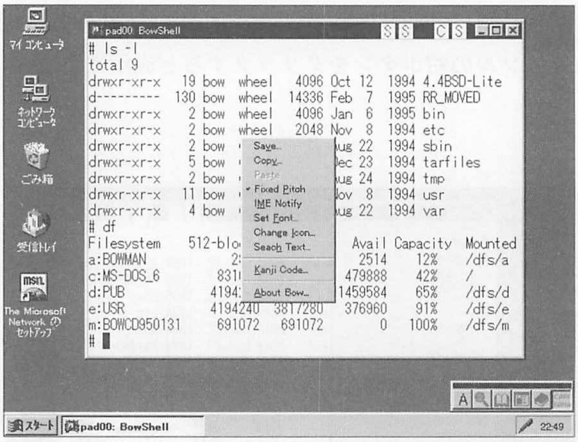


図 4-4　マウスのメニュー（Windows 95 の場合）

■フォントの選択

BOWPAD に表示されるフォントの選択はコントロールメニューの［Set Font...］を選択すると表示される、［フォントの指定］ダイアログボックス（図 4-5）で行います。最後に変更した設定が、新しいウィンドウを開くときの初期設定となります。

コントロールメニューの［Fixed Pitch］をチェックすると選択したフォントが固定幅だけになります。［Set Font...］でプロポーショナルフォントを選択しても文字幅は固定で表示されます。また、フォントサイズ、ウィンドウサイズの変更によって表示可能な文字数が変わった場合には SIGWINCH シグナル（端末の表示文字数が変更になったことを通知

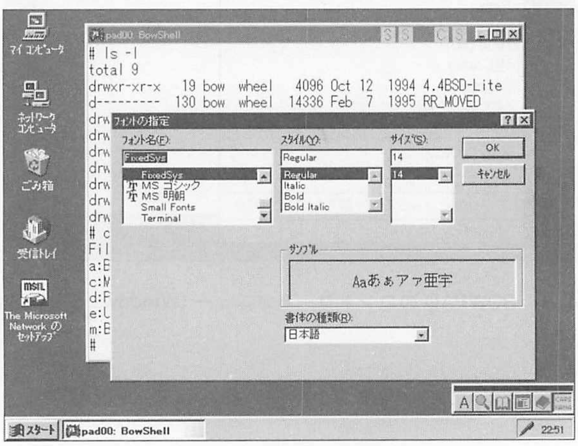


図 4-5　フォント設定ダイアログボックス

するシグナル)がプログラムに送られることによって、BSD プログラムが認識している端末の行数と桁数も自動的に変更されます。

■ クリップボードへのコピー／ペースト

コントロールメニューの[Copy]を選択すると[copy]ダイアログボックス(図 4-6)が表示されます。テキストボックス上のすべて、または一部を選択し、[Copy]ボタンでクリップボードにデータを送信します。テキストを選択せずに[Copy]ボタンをクリックした場合はすべてのデータが送信されます。

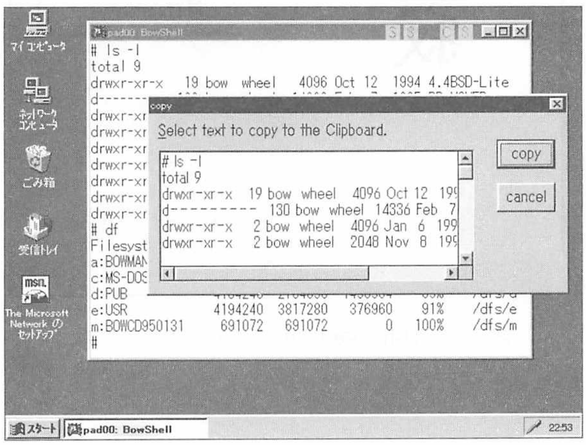


図 4-6 [copy]ダイアログボックス

クリップボード上のテキストデータを受信するにはコントロールメニューの[Paste]ボタンを選択します。

■ アイコンの設定

BOWPAD をタスクバー(Windows 3.1 の場合はデスクトップ上)にアイコン化したときのアイコンを設定します。この機能は BSD プログラムをプログラムマネージャにアイコン登録して(BOWPAD のシェル上からではなく)直接起動するときに、そのプログラムに合ったアイコンを設定することができます。

次に手順を説明します。コントロールメニューから[Change Icon...]を選択すると[Change Icon]ダイアログボックス(図 4-7)が表示されます。リスト表示されたアイコンをクリックするか、[Browse...]ボタンで表示されるダイアログボックからアイコンファイルをリストに呼び出してクリックします。[OK]ボタンをクリックすることによりアイコンの設

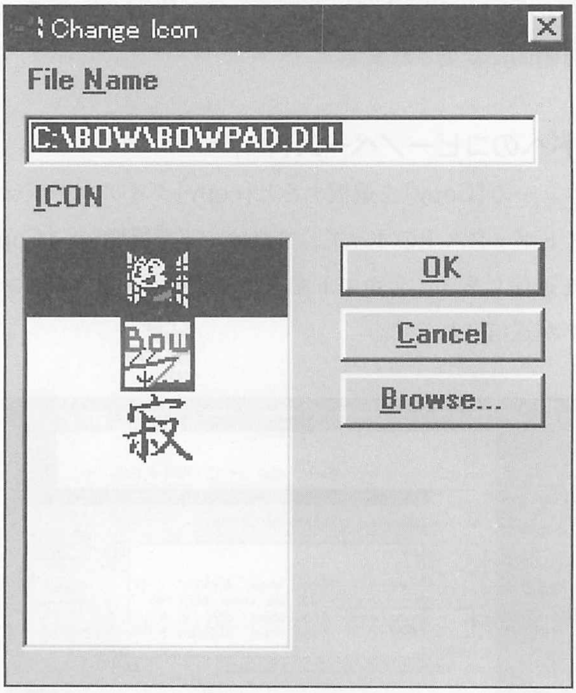


図4-7　[Change Icon]ダイアログボックス

定が完了します。

ここで設定した内容がプログラムマネージャやスタートメニューのアイコンに反映されないことに注意してください。

■ BOWPAD内の文字列の検索

BOWPADがCANONICALモードにあるとき、BOWPAD内に表示された文字列を、バックスクロールできる分まで含めて検索することができます。コントロールメニューから[Search Text]を選択すると、図4-8のようなダイアログボックスが表示されます。ここで、検索したい文字列を正規表現で入力し、検索の方向を指定して[次を検索]ボタンをクリックすると、正規表現にマッチした文字列の先頭にカーソルが移動します。

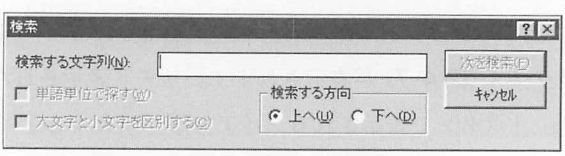


図4-8　文字列検索のダイアログボックス

■ 漢字コードの設定

BOW.INI の［BowPad］セクションの CClib エントリが JCONV.DLL に設定されているときは、キーボードから入力される漢字コード（Keyboard Code）と BOWPAD 上に表示される漢字コード（Display Code）を変更することができます。ここで設定した内容はタイトルバーのモード表示（77 ページを参照）に反映されます。

次に手順を説明します。コントロールメニューから［Kanji Code...］を選択すると漢字コードを設定するためのダイアログボックスが表示されます。

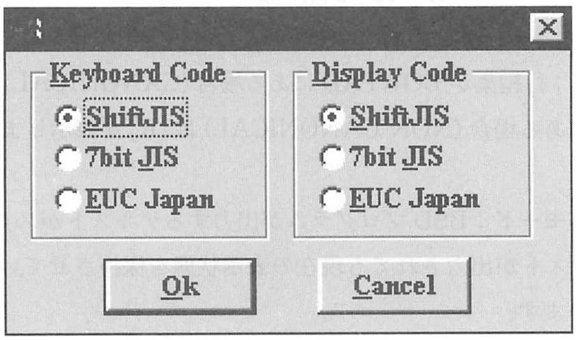


図 4-9　漢字コード変更ダイアログボックス

選択できる漢字コードはシフト JIS、JIS（7bit JIS または JUNET コード）、EUC の 3 種類です。［Keyboard Code］、［Display Code］に利用する BSD プログラムに合わせてコードの設定を行ってください。設定はプログラムマネージャに登録されているアイコンの［アイコンの登録内容の変更］［コマンドライン（C）］テキストボックスの BOW.EXE のコマンドラインに「-n」で指定されているプロセス名ごとに管理されています。同じ名前で起動された BOWPAD にも同じ漢字コードが設定されています。

■ モード表示

BOWPAD のタイトルバーには端末のモードが表示されます。

1. **キーボード用漢字コード**：キーボードから入力される漢字コードを表示します。［J］が JIS、［S］がシフト JIS、［E］が EUC を表します。
2. **表示用漢字コード**：BOWPAD に出力（表示）される漢字コード種別を表示します。表示内容は 1 と同じです。
3. **termcap モード**：termcap を利用した BSD プログラムを起動するとこの位置に［T］と表示されます。

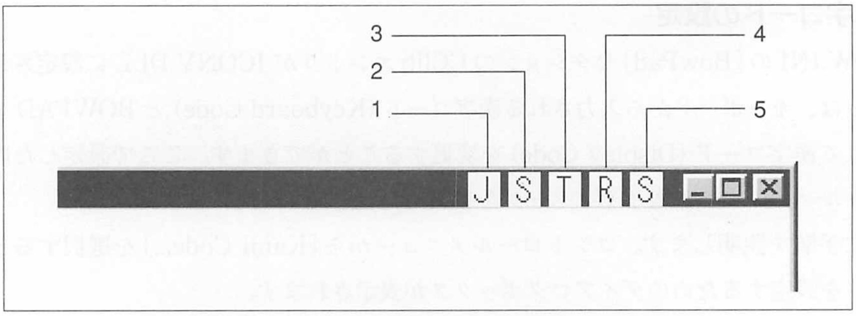


図 4-10　BOWPAD のモード表示

4. **編集モード**：行編集が BOWPAD にある場合 (CANONICAL) は [C] を、BSD プログラムにある場合 (NON CANONICAL) は [R] を表示します (190 ページを参照)。
5. **スクロールモード**：BSD プログラムが出力するテキストがスクロールする場合は [S]、テキストが出力されても現在の表示状態を保持させているときは [H] が表示されています。

■ コンソールの操作

NON CANONICAL モード (CBREAK および RAW) は emacs や vi、tcsh などのプログラムが使用するコンソール (端末) のモードです。このモードでは 191 ページの表で示されているキャラクタコードが BSD プログラムに渡されます。このモードでの端末の動作 (カーソル移動や行編集の操作など) は BSD プログラムが行なうので、操作法は各々のプログラムのマニュアルを参照してください。また、タイトルバーの編集モード表示は [R] になります。

一方の CANONICAL モードは、sh や csh など行編集を端末ドライバに任せているプログラムを使用している場合に使われます。

CANONICAL モード (行編集モード) では、キー入力は入力エリアで行ないます。入力エリアは Windows のエディットコントロール (各種ダイアログボックスのテキストボックスなど) と同等の行編集を行なうことができます。また、BSD の端末ドライバと同等のシグナル発生の機能もサポートしています。ただし、エディットコントロールでクリップボードへのコピーを行なう Ctrl ＋ C は、BSD では通常インタラプトシグナルを発生する文字として設定されているので使用することはできません*4。CANONICAL モードで

＊4　stty(1) などで intr を別の文字に変更すれば可能です。

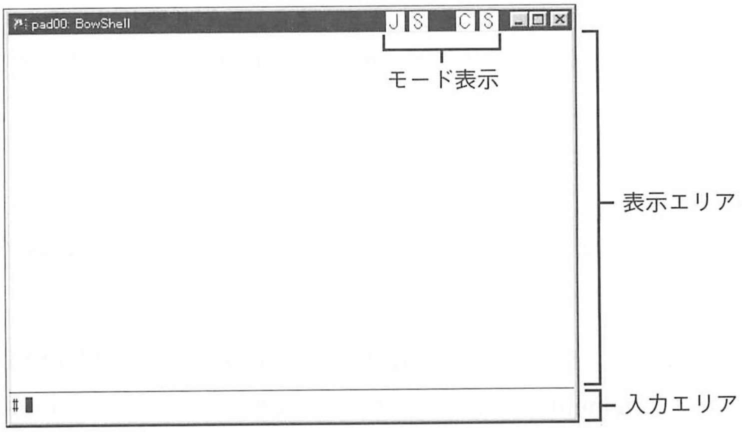


図4-11　スクリーンの構成

Ctrl ＋ C を入力すると、クリップボードへのコピーではなく、BOWPADを使用しているプロセスに対するインタラプトシグナルの送信として動作します。

表示エリアでは縦、横のスクロール、入力エリアへのコピーを行うことができます。表示エリアで機能するキーは表4-2のようになります。

表4-2　表示エリアで機能するキー

キー	動作
Scroll Lock (NEC PC-9800シリーズでは vf・5)	表示エリアのスクロールの停止／再開を切り替える。タイトルバーの一番右のモード表示に、停止中は[H]を再開すると[S]を表示する。
Insert	カーソル以降の文字を入力エリアにコピーする。
Page Up 、 Alt ＋ ↑	半ページ上へスクロール
Page Down 、 Alt ＋ ↓	半ページ下へスクロール
Alt ＋ ←	1/3ページ左へスクロール
Alt ＋ →	1/3ページ右へスクロール
Ctrl ＋ ↑	1行上へスクロール
Ctrl ＋ ↓	1行下へスクロール
Ctrl ＋ ←	1文字左へスクロール
Ctrl ＋ →	1文字右へスクロール
Alt ＋ Home	スクロールバッファの先頭へジャンプ
Alt ＋ End	スクロールバッファの最後へジャンプ
Home	行の先頭へジャンプ
End	行の最後へジャンプ

表 4-2以外のキーは入力エリアに送られます。また、CANONICAL モードのときはマウスが有効になり、左ボタンのクリックでカーソルの移動、ダブルクリックでカーソル以降の文字を入力エリアにコピーすることができます。

■ Telnet

Telnet プロトコルを使って遠隔ホストにログインする場合は次のことに気を付けてください。

1. termcap のエントリは/usr/share/misc/termcap の bowpad を使用します。
2. リモートの telnetd がウィンドウサイズの変更に対応しているならば BOW 上の telnet とリモートの telnetd との間でサイズ変更が自動的に通知されます。そうでない場合 (SunOS 4.X.X の telnetd など) は X11 に含まれる resize(1) コマンドでウィンドウサイズを再設定します。
3. terminfo を使用している System V 系のマシンでは、captoinfo(1) および tic(1) を使用して bowpad エントリを含む terminfo データベースを生成する必要があります。captoinfo(1) および tic(1) は System V のコマンドで BOW には含まれません。terminfo データベースの作成方法については、ログイン先のシステムのマニュアルを参照してください。

4.2.2 BOW.EXE

BOW.EXE はコマンドラインオプションに指定された BSD の a.out*5形式のバイナリファイルを実行します。BSD コマンドを実行する場合、内部で必ず BOW.EXE が実行されます。通常はユーザーが BOW.EXE を実行する必要はありませんが、プログラムマネージャやスタートメニューのアイコンに、シェル (sh、csh など) や emacs などの対話型プログラムを登録するときに必要になります (61 ページを参照) 。

■ コマンドラインオプション

BOW.EXE のコマンドラインオプション (以降はオプションと略します) は 2 種類に分類されます。a.out ファイル名より前は BOW.EXE に対するオプションです。a.out ファイル名以降のオプションは BSD プログラムを起動するシステムコール execve(2) にそのま

*5　本マニュアルでは BSD プログラムのバイナリイメージを指す一般名称として「a.out」を使用します。a.out は BSD 上の C コンパイラが生成するバイナリイメージの初期値となっています。

ま渡されます。そのためファイルの指定などは BSD 形式（ディレクトリの区切りが〝/〞）で与える必要があります。

コマンドラインの一般的な形式は次のようになります。

BOW.EXE [-n プロセス名] [-a] [-q] [-gXxY] a.out ファイル名 [argv[0] argv[1] ...]

-a	オートクローズ。BSD プログラムの終了と同時に BOWPAD も終了してウィンドウを閉じます。指定しないと、BSD プログラム終了後にも BOWPAD のウィンドウは閉じません。
-n プロセス名	プロセス名の指定。プロセス名は BOWPAD のタイトルバーおよび BOWPS の表示に使用されます。-n オプションを指定しないか、プロセス名を省略（「-n ""」と指定）したときは、「BowPad」という文字列に起動した端末の番号を付け加えた名前がプロセス名として使われます。たとえば、3 番目に起動された BOWPAD は「BowPad03」という名前になります。さらに argv[0]*6 も省略された場合は a.out ファイル名がプロセス名になります。
-q	quiet モード。BOWPAD のウィンドウは開かれず、標準入出力とエラー出力は/dev/null に割り当てられてコマンドが実行されます。バックグラウンドで処理を行ないたい場合やデーモンプログラム*7を実行したい場合などに使用します。
-gXxY	起動時のウィンドウサイズの指定。X に桁数を Y に行数を指定します。たとえば、80 桁で 25 行の場合は「-g80x25」と指定します。

BSD プログラムの中には argv[0] の内容を見て動作を変えるものがあります。

たとえば、sh、csh は argv[0] の先頭の文字が〝-〞の場合にはログインシェルであると判断し、sh の場合は「.profile」、csh の場合は「.login」を実行します。一般の BSD システムではこれを/bin/login が行ないますが、BOW では/bin/login を使用しないため argv[0] から指定するようになっています。

■ 環境変数

BSD プログラムは通常はシェルから起動されるために、環境変数は.profile、.login などのシェルの初期化ファイルで設定、管理を行ないます。BOW ではシェルからの起動以外にプログラムマネージャからの直接の起動もサポートしています。この場合は初期化ファイルが処理されませんので、あらかじめ設定しておきたい環境変数を BOW.INI の [Environments] セクションで指定しておくことができます。設定形式は次の通りです。

環境変数名＝設定内容

---

*6　argv は C 言語というプログラミング言語で、コマンドラインオプションを表す変数として一般的に用いられています。argv[0]、argv[1] と書くことで 0 番目（通常は実行ファイル名）、1 番目のオプションを表します。

*7　かな漢字変換サーバのように、直接ユーザーとのやりとりは行なわないプログラムの総称。DOS における TSR（常駐）プログラムに相当するもの。

4.2.3 BOWPS.EXE

[ps] アイコンとして登録されている BOWPS.EXE は、BOW のプロセスの一覧表示とシグナルの送信を行う Windows アプリケーションです。BOW では ps(1) と kill(1) の役割を BOWPS が果たします。プロセス一覧は 10 秒毎、またはコントロールメニューの [Refresh] の設定時間ごとに更新されます。BOW 1.5 では、さらにプロセスの詳細情報（メモリマップ、レジスタの値など）やデバッグコンソールを表示できるようになっています。

BOWPS は次のような内容を表示しています。

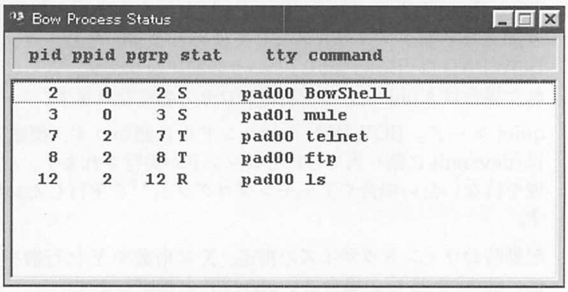


図 4-12　BOWPS

pid	プロセス ID
ppid	親プロセスの ID
pgrp	プロセスグループ ID
stat	プロセスの実行状態
tty	プロセスを起動した端末
command	実行中のコマンド

stat に示される文字の意味は次のとおりです。

R	実行可能なプロセス
T	停止したプロセス
S	スリープをしているプロセス
I	プロセス初期化中
Z	終了待ちプロセス

command には実行されているプログラムの種類によって次のように表示されます。

BSD コマンド	コマンド名
DOS アプリケーション	WINOLDAP
Windows アプリケーション	拡張子を省いた実行ファイル名

3 種類のプログラムのうち、DOS アプリケーションと Windows アプリケーションは疑似プロセスであるためプロセスにシグナルを送ることはできません。

プロセスにシグナルを送るには、プロセスリストで目的のプロセスを選択し、マウスでダブルクリック、Enter キー入力またはコントロールメニューの [kill process...] を選択す

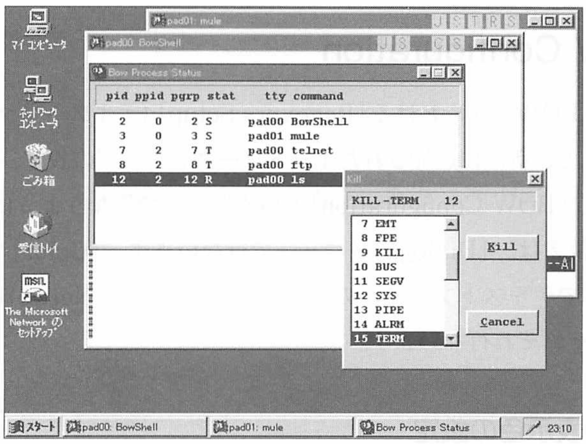


図 4-13　[Kill] ダイアログボックス

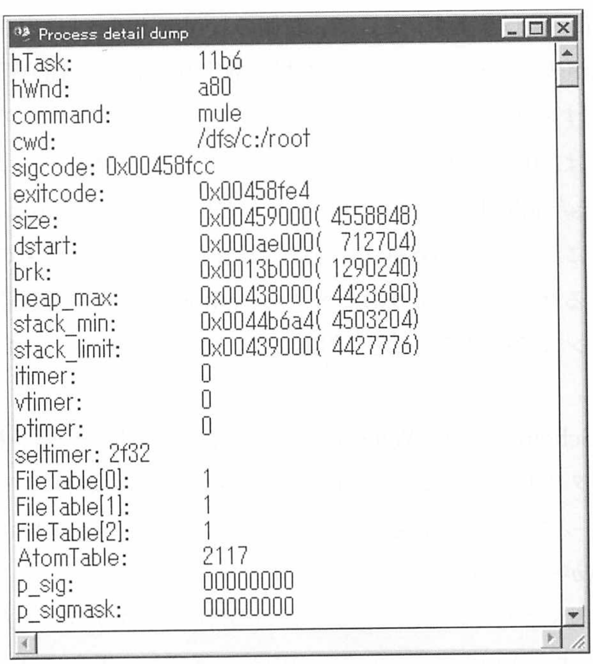


図 4-14　プロセスの詳細情報

ることでダイアログボックスを表示します。[kill]ダイアログボックス（図4-13）が表示されたら、リストボックスからシグナルを選択し[kill]ボタンをクリックしてシグナルを送信します。初期値はTERM（ソフトウェア終了シグナル）です。

プロセスの詳細情報を表示させたい場合は、目的のプロセスをクリックして選択し、キーボードから Ctrl ＋ D を入力します。

4.2.4 BOW Configuration

BOW 1.5では、BOW 1.0における[Regexprs Setup]（テキストファイルとして取り扱うファイルの指定）と、新たに追加されたネットワークドライブの設定、およびBOWPADの表示色の設定を、[BOW Configuration]というツールで行なうようになっています。設定する項目の切り替えは、[Dialogs]メニューで行ないます。[Color]でBOWPADの表示色、[Regexpr]でテキストファイルのファイル名の設定、[Network]でネットワークドライブの設定をそれぞれ行ないます。

■BOWPADの表示色の設定

BOW 1.5では、BOWPAD上に表示される色を、Windowsの色設定と別に設定することができます。BOW Configurarionの[Dialogs]メニューで[Color]を選択すると、図4-15のように表示が変わります。ここで、左側の[color index]は現在の色の設定状態を示しています。色が表示されているバーの右横の数字（0～7）は、BOWPADのエスケープシーケンスで指定する色番号の下一桁に相当しています。BSDプログラムで通常使われているものとしては、0番がウィンドウ中の通常の文字列、7番がウィンドウの背景、1番が反転表示（emacs/mule/lessのモードラインなど）の文字、6番が反転表示の背景です。そのほかの色番号については、BSDアプリケーションによって使うものが変わってきますが、実際に使われるケースはあまりありません。color indexの色番号と、BOWPADのエスケープシーケンスで指定できる色の対応については、189ページの表「**psの設定値**」を参照してください。

右側の[Color Scheme]では、Windowsの色設定で設定している各種の色（メニューの色、ウィンドウのタイトルバーの色など）と同じ色を使いたい場合に、その設定の名前を選択できるようになっています。中央より下のボックスには、現在選択されている色番号に設定されている色がそのまま表示されています。

任意の色番号に設定されている色を変更したい場合には、まず変更したい色番号を[color index]で選択します。Windowsの色設定の中から選びたい場合には、[Color Scheme]

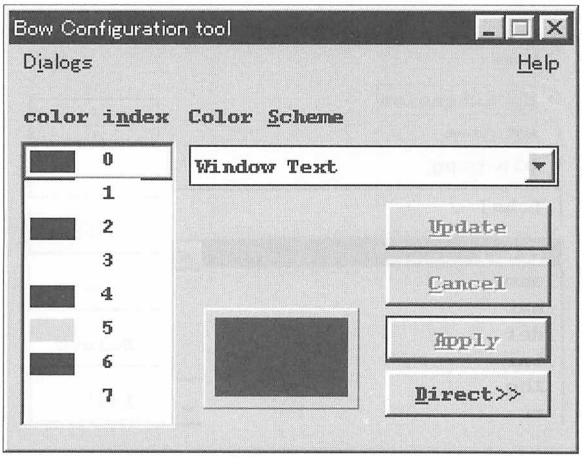


図4-15 BOWPADの表示色の設定

のドロップダウンリストの中から目的のものを選択します。直接色を作成したり、カラーリストから選択したい場合には、[Direct >>]ボタンをクリックし、表示されるダイアログボックス[色の設定]で目的の色を選択するか、新しく色を作成します。色の作成や選択が終わったら、[Apply]ボタンを押すと、左側の[color index]の内容が更新されます。色の設定が終わったら[Update]ボタンを押すと、変更の結果がBOW.INIの内容に反映されます。[Update]ボタンを押す前に[Cancel]ボタンを押せば、変更を取り消すことができます。

■テキストファイルのファイル名設定

[Dialogs]メニューの[Regexpr]を選択すると、BSDとDOSのファイルシステムの違い、すなわちテキストファイルにおける改行コードの違いやEOFへの対応が必要になるファイルの種類を設定することができます。この設定は、ファイル形式の変換が必要となるファイルの名前を、正規表現で指定することで行います。設定することができる項目は次の3つです。

by extension	拡張子によるテキストファイルの指定
by name	拡張子を除くファイル名によるテキストファイルの指定
bin prog	バイナリモードでファイルをオープンするプログラムの指定

たとえば、BOWを使う環境では、通常のMS-DOS、Windowsアプリケーションによって作成されたテキストファイルとemacsなどのエディタが作成したテキストファイルなど、改行コードの異なるファイルが混在することになります。それぞれのファイルをそれ

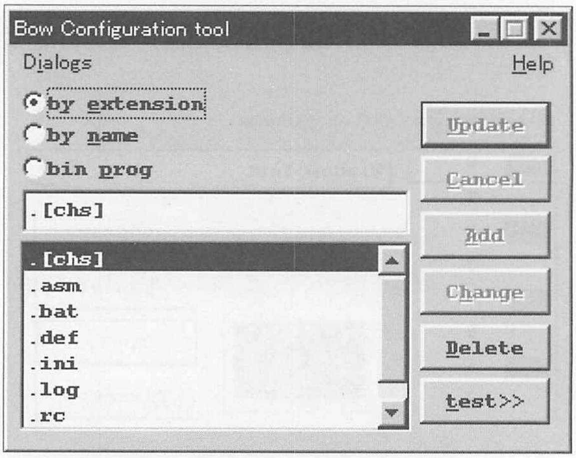


図 4-16　[Dialogs] [Regexpr] のウィンドウ

ぞれのアプリケーションが使っていれば問題ないのですが、BOW 上の emacs を使って、MS-DOS のバッチファイルを編集したりする場合には、改行コードを変換してやらないと面倒なことになります。そこで、BOW では、「このファイルをオープンするときにはテキストファイルだから改行コードを CR/LF から LF に変換するして BSD プログラムに渡す」という設定を行なえるようにしてあります。逆に、特定の BSD コマンドがファイルをオープンするときには、かならずバイナリモードにするという設定もできます。

テキストファイルとして取り扱いたいファイル名の指定は、次のように行ないます。まず、拡張子毎に設定を行ないたい場合には、

1. [by extension] のボタンを選択する。
2. リストに表示されている拡張子一覧の中に目的の拡張子が含まれていない場合、テキストボックスにその拡張子を正規表現で入力する。たとえば、.html あるいは.htm という拡張子をもつファイルをテキストファイルとして扱いたい場合には、

   ```
   .html?
   ```

   のように指定します (ここで「?」は直前の 1 文字が 0 または 1 回現われることを示す正規表現)。
3. [Add] ボタンを押すと、入力した正規表現がリストに追加される。すでにリストにある正規表現を選択して変更した場合には、[Change] ボタンを押す。
4. [test]ボタンを押すと、図 4-17のダイアログボックスが表示されるので、[program name] のテキストボックスにこのファイルをテキストファイルとして扱わせたい

適当なBSDプログラムの名前（たとえば「cat」）を、[/..]のテキストボックスにはテストしたいファイル名（「test.htm」、「test.html」、「test.htmla」などいろいろ試してみる）を入力し、[test]ボタンを押す。一番下の表示行に「Text File」と表示されれば、その名前のファイルはテキストファイルとして扱われる。「Binary File」と表示されれば、テキストファイルとしての変換は行なわれない。

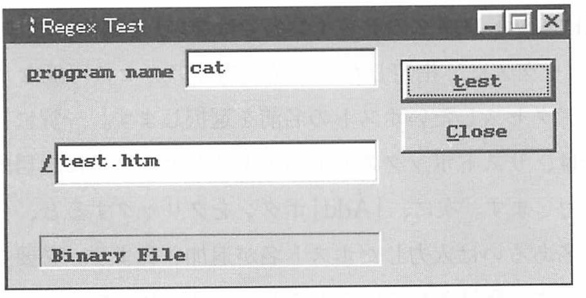


図 4-17　正規表現のテスト

次に、拡張子ではなく、ファイル名そのもので指定したい場合（たとえば、拡張子を持たない各種設定ファイルなど）の場合には、[by name]を選択して、同様にテキストボックスに目的のファイル名を正規表現で入力します。たとえば、「readme.1st」というファイルをテキストファイルとして処理したい場合（「.1st」という拡張子をもつものすべてではなく）には、

```
readme＼.1st
```

と入力します（"."（ピリオド）の前の"＼"は、"."の特殊な意味を打ち消すために必要です。日本語キーボードからは半角の"¥"で入力されます）。

特定のBSDコマンドで、テキストファイルの変換を行なわせたくない場合には、[bin prog]を選択して、テキストボックスのそのコマンド名を入力します。以後のテストとBOW.INIへの反映は前述のとおりです。

参考までに、BOWは、以下のロジックに従ってオープンするファイルの形式を判断しています。

1. [bin prog]で指定されたコマンドと一致した → バイナリでオープン
2. [by name]で指定されたファイル名と一致した → テキストでオープン
3. [by extension]で指定された拡張子と一致した → テキストでオープン
4. どれとも不一致 → バイナリでオープン

■ネットワークドライブの設定

BOW 1.5では、Microsoftネットワークの環境下で、リモートホストの共有ドライブにBSDプログラムからアクセスすることができます。BOW Configurationの[Dialogs]メニューの[Network]を選択すると、共有ドライブを、/dfs/[ホスト名]/[共有名]の形式でアクセスできるようになります。

まず、[Domain]のリストボックスから、アクセスしたいMicrosoftネットワークのドメイン名(TCP/IPネットワークのドメイン名ではない)を選択します。しばらく待つと、[Host]のリストボックスに、指定したドメイン内でアクセス可能なホストの一覧表が表示されますので、アクセスしたいホストの名前を選択します。一覧に目的のホストが表示されていない場合は、リストボックスの上のテキストボックスに、目的のホスト名を直接キーボードから入力します。次に、[Add]ボタンをクリックすると、[/dfs]のボックス内に選択したホスト名あるいは入力したホスト名が追加されます。必要なホストの分だけこの作業を繰り返します。登録されているホストへのアクセスを止めたい場合には、[/dfs]のリストボックスで登録を取り消したいホスト名を選択して、[Delete]ボタンをクリックします。[Update]ボタンをクリックすると、登録内容がBOW.INIに反映されます。

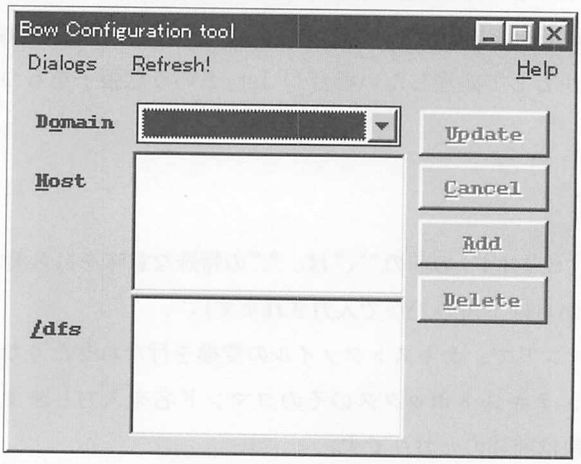


図4-18　ネットワークダイアログ

4.2.5 bsd2dos.exe、dos2bsd.exe

bsd2dos.exe は、引数で与えられた BSD ファイル名を MS-DOS ファイル名に変換して標準出力に書き出します。dos2bsd.exe は MS-DOS ファイル名を BSD ファイル名へ変換します dos2bsd.exe にはフルパスを渡さなければなりません。Windows 95 で長いファイル名が使える場合には変換は行なわれません。

このプログラムは BSD プロセスの子プロセスとして起動する必要があります。プログラムマネージャ、ファイルマネージャからの直接の起動はできません。直接起動する必要がある場合は、スクリプトを作成し、/bin/sh から実行すれば可能です。BOW.EXE のコマンドラインは次のようにしてします。

```
bow.exe /bin/sh sh スクリプト オプション
```

4.2.6 execvp

BOW から DOS、Windows アプリケーションをアイコン化して実行する際に使う、BOW 独自の BSD コマンドです。コマンドラインは次のとおりです。

execvp [-] 実行ファイル名 [-]argv[0] ...

argv[0] が〝-〟で始まるときはアイコン化した状態で DOS、Windows アプリケーションを起動します。実行ファイル名の前に〝-〟が指定されたときは、DOS、Windows アプリケーションは BOW の管理から切り離されます (BOWPS のプロセス一覧に表示されない)。

4.2.7 crpad

新規の BOWPAD を開き、標準入力を書き込む BOW 独自の BSD コマンドです。more などのページャの代りに使用できます。 コマンドラインは次の通りです。

crpad [-e コマンド]

-e オプションで別の BSD コマンドを指定すると、新しく開いた BOWPAD 上のデータを処理することも可能です。

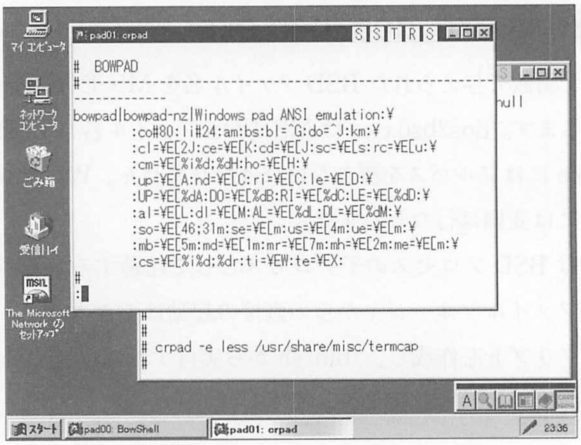


図 4-19　crpad

4.2.8　manbow

manbow*8は BSD のオンラインマニュアルを Windows 3.1 のヘルプファイルへ変換します。利用するためには別途 Windows SDK などに含まれる HC31.EXE が必要です。

使用法は manbow ディレクトリの下に man[1-8] のディレクトリを作成し、BSD のマニュアルのソース（nroff でフォーマットする前のファイル）をそれぞれの下に置き、次の形式で変換します。

```
manbow <変換するセクション番号>
```

次の例はセクション 1、2、3 のマニュアルを作成します。

```
manbow 1 2 3
```

＊8　編注：manbow は作者である太田博志氏がパブリックドメインとし、すべての権利と責任を放棄しています。

4.3 BOW.INI

BOW.INI は BOW の環境設定ファイルです。基本的には BOW をセットアップした状態で問題ありませんが、BOW を使いやすくするためには変更が必要になる場合があります。

BOWPAD 上のシェルからではなく、直接 BSD プログラムを起動する場合は、そのプログラムが参照する環境変数をあらかじめ BOW.INI に指定しておく必要があります。

4.3.1 [BowPs]セクション

RefreshRate＝整数

BOWPS で表示されるプロセス一覧の更新時間間隔 (秒) を指定します。デフォルトは 10(秒) です。

4.3.2 [init]セクション

DriveFileSystem＝/dfs

変更しないでください。

Console＝/dev/pad%02d

BOW.EXE が Windows シェルから起動されたとき、標準入出力になるデバイスを指定します。%02d の部分を 00 から [init] セクション nPads で指定された値-1 まで、排他オープンに成功するまでオープンを試みます。

PadName＝/dev/pad%02d

変更しないでください。

nPads＝整数

[init] セクション Console の上限値を指定します。これは BOWPAD.DLL によって書き込まれます。デフォルトは 32 です。

RootDrive＝ドライブレター

ルート (/) になるドライブを指定します。デフォルトは c ドライブです。

OsType＝OS タイプ

uname -s で返される値を指定します。デフォルトは 4.4BSD です。

OsRelease＝OS リリース

uname -r で返される値を指定します。デフォルトは 4.4BSD-BOW です。

OsVersion＝OSバージョン

uname -vで返される値を指定します。デフォルトはBOWの内部バージョン、タイムスタンプです。

4.3.3 ［Environments］セクション

環境変数名＝値

BSDのプロセスに渡す環境変数を、「環境変数名＝値」の形式で設定します。HOME、TZ、HOSTNAMEはBOWのカーネルでも参照しているので必ず指定します。HOMEは/etc/passwdの指定と合わせておきます。［BSD setup］アイコンの［Passwd DB］で設定します。ここで設定した内容はBOWのプロセスをすべて終了しなければ反映されません。

4.3.4 ［BowPad］セクション

PadBehavior＝［ansi|domain］

CANONICALモードでのINPUTPADの振る舞いを指定します。デフォルトはansiです。DOMAIN風にするにはdomainを指定します。

Window名 Flags＝整数値

BOWPADのフラグが記されています。終了時の記憶用です。

Window名 ICON＝ファイル名,ICON番号

アイコン化したときのアイコンが記されています。終了時の記憶用です。

Window名 Size＝ファイル名,ICON番号

Windowサイズが記されています。終了時の記憶用です。

Window名 Font＝Logical Font

フォントが記されてます。終了時の記憶用です。

CCLib＝DLL名

METAキーの修飾を行なうDLLを指定します。デフォルトはJCONV.DLLです。JCONV.DLLはALTキーでMETA修飾を行ない、キー入力の漢字コードを変換します。NOCONV.DLLはMETA修飾を行ないますが、キー入力の漢字コードを変換しません。META修飾、漢字コード変換が必要ない場合は無指定でも構いません（この行自体が必要ありません）。emacs（英語版）を使うときはNO

CONV.DLL を、Nemacs または Mule を使うときは JCONV.DLL を使い、さらに Nemacs、Mule および BOWPAD の keyboard コードを 7bit jis(junet) に設定すれば、META-KEY と IME のモード切り替えを同時に使用することが可能です。1 バイトカナの入力には対応していません。

ColorMap＝カラーマップ

カラーマップを指定します。BOW.INI を直接変更せずに BOW Configure (BOWCONF.EXE) で設定してください。

4.3.5　[debug]セクション

Int24Ignore＝[file]

file を指定すると、ディスクが入っていないフロッピードライブにアクセスしたときの警告ダイアログボックスの表示を抑止します。

Inquire HeapSize＝0|1

Inquire StackSize＝0|1

Inquire MMapSize＝0|1

それぞれ、[HeapSize],[StackSize],[MMapSize] セクションに設定のないプログラムをロードしようとしたときに、0 以外の値が指定してあるときにはユーザに問い合わせます。

UnsupportedSyscall＝Query|Ignore

BOW が実装していないシステムコールが発行されたときの動作を指定します。Query を指定すると ENOSYS を返すか、終了するかダイアログボックスで選択するようにします。Ignore を指定すると無条件に BSD プロセスに ENOSYS を返します。デフォルトは Query です。

DefaultArgs＝options a.out argv[0] ...

BOW.EXE が引数なしで起動されたときのデフォルトの引数を指定します。デフォルトは/bin/sh -sh です。

VforkDataCopy＝yes|no

vfork(2) システムコールの際にデータ領域をコピーする/しないを指定します。no を指定するとプロセスの生成は高速になりますが、vfork された子プロセスがデータ領域を書き換えるような、vfork を正しく使用していないバイナリを実行するときには誤動作するので yes を指定します (このようなバイナリは vfork の使用法に

違反していますが、BSDのオリジナルライブラリのexecvp等がこれを行います)。BSDプログラムCD-ROMからインストールしたバイナリまたはBSDプログラムCD-ROMのライブラリを使ってコンパイルしたバイナリだけを使用する場合は、noでかまいません。デフォルトはyesです。

VforkAbuseApps＝a.out1,a.out2,...

vfork(2)を正しく使用していないプログラムを指定します。ここで指定されたプログラムはVforkDataCopy＝noの場合でもデータ領域をコピーします。

Trace＝トレース項目1, トレース項目2,...

以下のトレース項目を指定します。トレース情報はBOWPS.EXEで開かれるデバッグターミナルに出力されます。

syscall	システムコール
signal	シグナル
ioctl	ioctl
fopen	ファイルオープン
inet	Winsockトレース

LongNameOnly＝yes|no

ファイル名がロングファイルネームのみとしてアクセスする／しないを指定します。この値がnoの場合はロングファイルネームでアクセスできなければ、従来API(BOW1.0のAPI)でアクセスします。たとえば、「12345678.txt」というファイルがある場合には、open("123456789.txt")は、このエントリの値がnoの場合は成功、yesの場合は失敗することになります。このエントリの値はWindows95のみ意味をもちます。デフォルトはyesです。

UtoLTranslation＝yes|no

ファイル名の変換(大文字→小文字、特殊文字)を行なう／行なわないを指定します。このエントリの値はWindows95のみ意味を持ちます。デフォルトではnoです。

4.3.6　[Load]セクション

DLL名＝[optional,][query,][引数]

ロードするBOWドライバを指定します。引数はドライバに渡されます。optionalが指定された場合はロードが失敗してもBOWの起動を実行します。queryが指定された場合はロードの前に確認のダイアログボックスを表示します。

4.3.7　[HeapSize]セクション

a.out ファイル名＝整数値 [KM]

default＝整数値 [KM]

malloc(3) システムコールで確保可能なメモリ領域の最大（全体）サイズをバイト単位で指定します。ファイル名にはディレクトリを含めません。K バイト単位の場合は「整数値 K」、M バイト単位の場合は「整数値 M」のように指定します。「1.5M」のような指定はできません。

4.3.8　[StackSize]セクション

a.out ファイル名＝整数値 [KM]

default＝整数値 [KM]

スタックの最大サイズをバイト単位で指定します。ファイル名にはディレクトリを含めません。K バイト単位の場合は「整数値 K」、M バイト単位の場合は「整数値 M」のように指定します。「1.5M」のような指定はできません。

4.3.9　[MMapSize]セクション

a.out ファイル名＝整数値 [KM]

default＝整数値 [KM]

mmap でマップ可能な最大サイズを指定します。K バイト単位の場合は「整数値 K」、M バイト単位の場合は「整数値 M」のように指定します。「1.5M」のような指定はできません。1M バイトあたり (システム全体で 8192 個のうちの)16 個のセレクタを消費します。シェアードライブラリを使用しない場合は、デフォルトのままにしてください。巨大なサイズを指定するとメモリブロックのフラグメンテーションの原因になります。

4.3.10 [BowRFD]セクション

DiskParam[0-4]＝type,head,cylinder,sec/track,bytes/sec

raw floppy ドライバのパラメータを指定します。6 種類のフォーマットを登録できますがフォーマットは DOS のデバイスドライバでサポートされているものでなければなりません。

type	フォーマット形式
0	2D
1	2HC
2	2DD
4	2HD
7	1.44
9	2ED

パラメータ	内　容
head	ヘッド数
cylinder	シリンダ数
sec/track	1 トラックあたりのセクタ数
bytes/sec	1 セクタあたりのバイト数

5章

BSDコマンド入門

本章は、BSD 初心者のユーザーの便宜を図るために、代表的な BSD コマンドのマニュアルで構成されています。オプションの解説やサンプルオペレーションについては、必要最小限のものにとどめています。BSD コマンドの詳しい使い方については、市販の UNIX 入門書籍などを参照してください。

5.1 基本コマンド

5.1.1 cat(ファイルの連結と出力)

■書式

cat [-benstuv] [-] [file ...]

■機能

入力された各行を標準出力に書き出します。

■オプション

-b	-n オプションと同様に番号を付けますが、空行には番号をつけません。
-e	-v オプションと同様に非印字文字を表示しますがさらに行末に '$' を表示します。
-n	1から始まる行番号を付けます。
-s	連続した複数の空行を1行にします。
-t	-v オプションと同様に非印字文字を表示しますがさらにタブを '^I' と表示します。
-u	出力をバッファリングしません。
-v	非印字文字を見えるような形で出力します。

■サンプルオペレーション

・ファイルの内容を出力します。

```
$ cat file
```

・ファイルを連結します。

```
$ cat file1 file2
```

・非印字文字を表示します。

```
$ cat file
        ASCII
BSD on Windows


$ cat -vet file
^IA^HAS^HSC^HCI^HII^HI$
BSD on Windows$
^L$
```

5.1.2　cp（ファイルのコピー）

■書式

cp [-R [-H | -L | -P]] [-fip] source_file target_file

cp [-R [-H | -L | -P]] [-fip] source_file ... target_directory

■機能

コピー元のファイルをコピー先へコピーします。

■オプション

- -H　-R オプションと同時に指定されたときにはコマンドラインで指定されたシンボリックリンクに対しては参照されているファイルをコピーしますが、サブディレクトリ中のシンボリックリンクに関してはシンボリックリンク自体をコピーします。
- -L　-R オプションと同時に指定されたとき、シンボリックリンクはシンボリックリンクにより参照されたファイルをコピーします。
- -P　-R オプションと同時に指定されたとき、シンボリックリンクはシンボリックリンク自体をコピーします。そのためリンク先が存在しない可能性があります。
- -R　コピー元がディレクトリのときにはそのディレクトリ以下すべてのサブディレクトリとファイルをコピーします。
- -f　コピー先にすでにファイルが存在しているときには確認のプロンプトなしにファイルを削除し新しいファイルを作成します。
- -i　コピー先にすでにファイルが存在しているときには確認のプロンプトを表示します。
- -p　修正時刻、アクセス時刻、ファイルフラグ、許可モード、ユーザー ID、グループ ID を保持してコピーします。

■サンプルオペレーション

・ホームディレクトリ以下を/tmp/home にコピーします。

```
$ cp -R ~ /tmp/home
```

・/tmp/bow というファイルが存在するとき上書きの確認をします。

```
$ cp -i ~/bow /tmp/bow
overwrite bow? y
```

5.1.3 ls(ディレクトリの内容の表示)

■書式

ls [-ACFLRTacdfiloqrstu1] [file ...]

■機能

ディレクトリの内容を表示します。

■オプション

-A	'.' と '..' をのぞいたすべてのファイルを表示します。
-C	マルチカラム形式で出力します。
-F	ファイルがディレクトリーのときには/、実行可能のときには*、シンボリックリンクのときには@をそれぞれファイル名の後ろに表示します。
-L	引数がシンボリックリンクのとき、シンボリックリンクが参照しているファイルについて表示します。
-R	サブディレクトリーを再帰的に表示します。
-T	完全な時間の情報を表示します。
-a	'.' と '..' を含むすべてのファイルを表示します。
-c	出力のソートのとき最終修正時刻でソートします。
-d	引数がディレクトリのときディレクトリ自身について表示します。
-f	出力をソートしません。
-i	i ノード番号を表示します。
-k	-s オプションが指定されているときにはファイルサイズをキロバイト形式で出力します。
-l	長い形式で出力します。
-o	-l オプションが指定されているときにはファイルフラグも表示します。
-q	ファイル名に非印字文字が含まれているときその文字を '?' として表示します。
-r	ソートを逆順にします。
-s	ファイルサイズを実際に使用しているフロック数で表示します。
-t	辞書順でソートする前に最終修正時刻順にソートします。
-u	最終アクセス時刻順にソートします。
-1	1 行に 1 エントリで表示します。

■サンプルオペレーション

・-F オプションを付けると、ファイルとディレクトリの別などが表示されます。

```
$ ls -F
directory/  file1*    file2*    otherdir/    symboliclink@
```

5.1.4 mv(ファイルの移動)

■書式

mv [-f | -i] source target

mv [-f | -i] source ... source directory

■機能

ファイルとディレクトリを移動します。ファイルまたはディレクトリの名前を変更します。

■オプション

-f 上書きになるときに確認のプロンプトを表示しません。

-i 上書きになるときに確認のプロンプトを表示します。

■サンプルオペレーション

・bowというファイルが存在するとき確認のプロンプトを表示します。

```
$ mv -i ascii bow
overwrite bow? y
```

・デフォルトでは確認のプロンプトは表示されません。

```
$ mv bow bsd
```

5.1.5　rm（ファイルの削除）

■書式

rm [-][-fir] filename...

■機能

1個または複数のファイルを削除します。

■オプション

-d	ファイルと同様にディレクトリを削除します。
-f	ファイルのパーミッションに関わらず、確認のプロンプトなしに削除します。
-i	ファイルのパーミッションに関わらず、各々のファイルを削除する前に確認のプロンプトを表示します。
-P	ファイルに上書きしてから削除します。
-R	ディレクトリ以下のすべてのファイル、ディレクトリを再帰的に削除します。
-r	-Rオプションと同じです。

■サンプルオペレーション

・確認のプロンプトを表示します。

```
$ rm -i bow
remove bsd? y
```

・ディレクトリ tempdir 以下のすべてのファイル、サブディレクトリを強制的に削除します。

```
$ rm -rf tempdir
```

5.1.6　rmdir（ディレクトリの削除）

■書式

rmdir directory...

■機能

ディレクトリを削除します。ただし、ディレクトリの中は空である必要があります。

■オプション

なし

■サンプルオペレーション

・ディレクトリの中が空でないときには削除できません。

```
$ rmdir ascii/
rmdir: ascii/: Directory not empty
```

5.1.7 mkdir（ディレクトリの作成）

■書式

mkdir [-p] [-m mode] directory_name ...

■機能

ディレクトリを作成します。

■オプション

-m mode	作成するディレクトリのアクセスモードを設定します。
-p	必要であればパスの中間のディレクトリを作成します。

■サンプルオペレーション

・/tmp/bow/というディレクトリが存在しない場合、/tmp/bow/source/というディレクトリはオプションなしでは作成できません。

```
$ mkdir /tmp/bow/source
mkdir: cannot make directory '/tmp/bow/source': No suchfile or directory
```

このようなときには-p オプションを使います。

```
$ mkdir -p /tmp/bow/source
```

5.1.8　pwd（カレントディレクトリ名の表示）

■書式

pwd

■機能

現在のディレクトリーを絶対パス名で表示します。

■オプション

なし

■サンプルオペレーション

・カレントディレクトリを絶対パスで表示します。

```
$ pwd
/home/bow
```

5.2　シェル

5.2.1　bash（GNU Bourne-Again SHell）

■書式

bash [options] [file]

■機能

sh のコマンドと互換性があり、また Korn Shell と C shell の便利な面を取り入れた shell 環境を提供します。

■オプション

-c string	コマンドを string から読み込みます。
-i	対話モードに入ります。
-s	-s オプション以降はコマンドを標準入力から読み込みます。
-	-をオプションの終わりとして解釈します。
-norc	ユーザー別初期化ファイル~/.bashrc を読み込みません。
-noprofile	/etc/profile などいっさいの初期化ファイルを読み込みません。
-rcfile file	初期化ファイルとして~/.bashrc ファイルの代わりに file から読み込みます。
-version	バージョン番号を表示します。
-quiet	静粛モード。
-login	ログインシェルとして起動します。
-nobraceexpansion	brace expansion を無効にします。
-nolineediting	ラインエディティングを無効にします。

■コマンドラインでの操作

TAB	ファイル名、ユーザ名、ホスト名、コマンド名を補完します。
C-p	ヒストリのリストから前のコマンドを取り出します。
C-n	ヒストリのリストから次のコマンドを取り出します。
C-b	コマンドラインでカーソルを左に移動します。
C-f	コマンドラインでカーソルを右に移動します。
C-a	コマンドラインでカーソルを行頭に移動します。
C-e	コマンドラインでカーソルを行末に移動します。
C-d	コマンドラインでカーソルのある文字を削除します。
C-c	フォアグラウンドで実行されている job を強制終了します。
C-z	フォアグラウンドで実行されている job を一時停止させます。

■シェルコマンド

for name [in word;] do list ; done	wordを展開してシェル変数nameに一つずつ設定してlistを実行します。
select name [in word;] do list : done	wordを展開してシェル変数nameに一つずつ設定してlistを実行します。ただし、nameの値は設定される度に標準エラー出力に出力します。
case word in [pattern [| pattern] ...) list ;;] ... esac	wordに一致する最初のpatternに対応するlistを実行します。
if list1 then command1 [elif list2 then command2] ... [else command3] fi	list1の終了ステータスが0ならcommand1を、list1が0でなく、list2が0ならcommand2をのように実行。list?のいずれも0でなければ、command3を実行します。
while list1 do list2 done	list1の終了ステータスが0である限り、list2を実行します。
until list1 do list2 done	list1の終了ステータスが0で「ない」限り、list2を実行します。

5.2.2 csh（C言語のようなコマンド文法をもったシェル）

■書式

csh [-bcefimnstvVxX] [arg ...]

csh [-l]

■機能

ヒストリー、ジョブコントロール、対話的なファイル名やユーザ名の補完、C 言語のような文法を組み入れた shell 環境を提供します。

■オプション

-b	オプションの処理を強制的に中止します。それ以降のシェルへの引数はオプションではないとして扱われます。
-e	コマンドが以上終了またはゼロ以外の終了ステータスで終了したときにシェルも終了します。
-f	.cshrc を読み込まずに高速に起動します。
-i	標準入力が端末でなくても対話的モードに入ります。
-l	ログインシェルとして起動します。
-m	.cshrc を読み込んでその結果がユーザに反映されなかったとしても.cshrc を読み込みます。
-n	コマンドの文法チェックをしますが実行はしません。
-s	コマンドを標準入力から受け取ります。
-t	入力から 1 行だけ読み込んで実行します。
-v	冗長モード。verbose 変数をセットします。
-x	echo 変数をセットします。コマンドは実行の前に直ちにエコーされます。
-V	.cshrc を実行する前に verbose 変数をセットします。
-X	.cshrc を実行する前に echo 変数をセットします。

■コマンドラインでの操作

C-c	フォアグラウンドのジョブを強制終了します。
C-z	フォアグラウンドのジョブを一時停止します。

■コマンド

・alias name wordlist

コマンドのエイリアス（別名）を設定します。

・bg [%job]

指定されたジョブをバックグラウンドジョブにします。引数がない場合にはカレントジョブをバックグラウンドジョブにします。

・cd [name]

カレントディレクトリを name というディレクトリに移動します。引数がない場合にはホームディレクトリに移動します。

・echo [-n] wordlist

wordlistを標準出力に出力します。-nオプションをつけると改行文字を付加しません。

・fg [%job]

指定されたジョブをフォアグラウンドジョブにします。引数がない場合にはカレントジョブをフォアグラウンドジョブにします。

・foreach name (wordlist)

wordlistを次々と変数nameに設定しながら「end」と入力するまでのコマンド行を実行します。

・history [-r] [-h] [n]

n個のヒストリーを表示します。-rオプションは逆順にします。-hは番号を表示しません。

・if (expr) comamnd

exprが真のときcommandを実行します。

・if (expr) then ... else if (expr2) then ... else endif

cshのif構文です。if構文の最後には必ずendifが必要です。elseの部分はなくてもかまいません。

・jobs [-l]

バックグラウンドで実行中あるいは停止中のジョブを表示します。-lオプションをつけるとプロセスIDも表示します。

・kill %job

指定されたバックグラウンドまたは停止中のジョブを終了させます。

・kill pid

指定されたプロセスを終了させます。

・logout

ログインシェルを終了します。

・rehash

環境変数PATHで指定されたディレクトリの内部ハッシュテーブルを再構築します。新しいコマンドをインストールした場合に実行します。

・set

すべてのシェル変数を表示します。

・set name

シェル変数nameの値を削除します。

・set name＝word

シェル変数 name に値を設定します。

・setenv

すべての環境変数を表示します。

・setenv name

環境変数 name の値を削除します。

・setenv name vlaue

環境変数 name に値を設定します。

・stop [%job] ...

指定されたジョブを停止します。

・unalias pattern

pattern に一致するすべてのエイリアスを削除します。

・unset pattern

pattern に一致するすべてのシェル変数を削除します。

・unsetenv pattern

pattern に一致するすべての環境変数を削除します。

・which command

実行される command の実行ファイルの位置を絶対パス名で表示します。

・while (expr)...end

expr が 0 でない限り、while と end によってはさまれたコマンドを繰り返し実行します。

5.2.3　vi(Visual Editor)

■書式

ex [-eFRrsv] [-c cmd] [-t tag] [-w size] [file ...]

vi [-eFRrv] [-c cmd] [-t tag] [-w size] [file ...]

view [-eFRrv] [-c cmd] [-t tag] [-w size] [file ...]

■機能

テキストファイルを扱うスクリーンエディタです。

■オプション

-R	読みだし専用モードで起動します。
-r	システムクラッシュ後、最近の編集内容を復元します。
+	起動時に最終行からファイルをオープンします。
+n	起動時にn行目からファイルをオープンします。
+/pattern	起動時にpatternの位置からオープンします。

■コマンド

「^文字」は、Ctrl キーを押しながら、文字キーを押すことを意味します。

ESC	コマンドモードに戻る
:wq	保存して終了
:q	終了
:q!	強制終了
a	カーソルの後ろにテキストを追加します。
A	行末にテキストを追加します。
b	カーソルを前の単語の先頭に移動します。
B	カーソルを前の単語の先頭に移動します(ただし、句読点などは無視します)。
^B	1画面分、逆方向スクロールします。
c	変更コマンド。
C	カレント行の行末まで変更します。
d	削除コマンド。
D	カレント行の行末まで削除します。
^D	半画面分、順方向スクロールします。
e	カーソルを単語の末尾に移動します。
E	カーソルを単語の末尾に移動します(ただし、句読点などは無視します)。
^E	一行だけ、順方向スクロールします。
fx	その行にある次のxを見つけます。
Fx	その行にある前のxを見つけます。

^F	1画面分、順方向スクロールします。
G	指定の行、またはファイルの最後の行に移動します。
^G	ファイルについての情報を表示します。
h	カーソルを左に移動します。
H	画面の左上にカーソルを移動します。
^H	カーソルを左に移動します。入力モードではバックスペースに同じです。
i	カーソルの前に新しいテキストを挿入します。
I	行の先頭にテキストを挿入します。
^I	コマンドモードでは機能なし。入力モードではタブに同じです。
j	カーソルを下に移動します。
J	行を結合します。
^J	カーソルを下に移動します。
k	カーソルを上に移動します。
l	カーソルを右に移動します。
L	画面の左下にカーソルを移動します。
^L	画面を描き直します。
m	カーソルの現在位置をマークします。
M	画面の中央にカーソルを移動します。
^M	キャリッジリターン。
n	最後の検索コマンドを順方向に繰り返します。
N	最後の検索コマンドを逆方向に繰り返します。
^N	カーソルを下に移動します。
o	カーソル行の下に新しい行をオープンします。
O	カーソル行の上に新しい行をオープンします。
p	バッファからテキストを追加します。
P	バッファからテキストを挿入します。
^P	カーソルを上に移動します。
Q	vi モードを抜けて ex モードにスイッチします。
^Q	コマンドモードでは機能なし。入力モードでは、次の文字を解釈せずに入力します。
r	カーソル位置の文字をタイプする文字列に変更します。
R	複数の文字を置き換えます。
^R	画面を描き直します。
s	カーソル位置の文字をタイプする文字に変更します。
S	カレント行をテキストで置き換えます。
tx	カレント行で次にある x の前の文字にカーソルを移動します。
Tx	カレント行で前にある x の次の文字にカーソルを移動します。
^T	コマンドモードでは機能なし。入力モードでは autoindent オプションがセットされている場合に使われます。
u	最後の変更を取り消します。

U	変更を取り消してカレント行を復元します。
^U	半画面分、逆方向にスクロールします。
^V	コマンドモードでは機能なし。入力モードでは次の文字を解釈せずに入力します。
w	次の単語の先頭にカーソルを移動します。
W	次の単語の先頭にカーソルを移動します（ただし、句読点などは無視します）。
^W	コマンドモードでは機能なし。入力モードでは、単語の先頭まで戻ります。
x	カーソル位置の文字を削除します。
X	カーソルの前にある文字を削除します。
y	ヤンクコマンド。
Y	カレント行のコピーを作ります。
^Y	一行だけ逆方向スクロールします。
z	画面を書き直し、カーソルの位置の行を画面の中央、最上行、最下行のいずれかに移動します。
ZZ	ファイルを保存して vi を終了します。

5.3 フィルタなど

5.3.1 tr(文字の変換)

■書式

tr [-cs] string1 string2

tr [-c] -d string1

tr [-c] -s string1

tr [-c] -ds string1 string2

■機能

標準入力からの入力に対して選択された文字を置き換え、または削除して標準出力に出力します。

■オプション

-c	指定された文字の補集合に対して動作します。
-d	指定されたされた文字を削除します。
-s	指定された文字の並びを1文字に圧縮します。

■サンプルオペレーション

・大文字から小文字に変換します。

```
$ tr '[a-z]' '[A-Z]' < file1
```

・すべての数字を削除します。

```
$ tr -d '0123456789' < f1
```

5.3.2　pr（ファイル内容の整形表示）

■書式

pr [+page] [-column] [-adFmrt] [[-e] [char] [gap]] [-h header] [[-i] [char] [gap]] [-l lines] [-o offset] [[-s] [char]] [[-n] [char] [width]] [-w width] [-] [file ...]

■機能

テキストファイルを整形して標準出力に出力します。

■オプション

+page	page で指定されたページから出力します。
-column	縦の段数（カラム）を指定します。
-a	入力行を複数カラム化して出力します。
-d	行間を1行ずつ空けて出力します。
-e [char][gap]	タブ文字を n*gap+1 の位置まで展開します。char が指定されたときには char をタブ文字と見なします。
-F	改ページするときに改行文字ではなく、改ページ文字を用います。
-h header	ヘッダーを指定します。
-i [char][gap]	複数のスペースを n*gap+1 の位置までタブ文字に置き換えます。char が指定されたときにはタブ文字の代わりとして使用します。
-l lines	1ページの行数を指定します。デフォルトは66行です。
-m	複数のファイルを同時にカラム単位に配置して出力します。
-n [char][width]	width 桁の行番号を付けます。デフォルトでは行番号のあとにはタブ文字が出力されますが、char が指定されたときには行番号のあとに char が出力されます。
-o offset	各々の行に offset 個のスペースがオフセットとして付加されます。
-r	ファイルのオープンに失敗しても診断的報告を出力しません。
-s char	カラムを区切る文字を指定します。デフォルトはタブです。
-t	通常5行ずつあるヘッダとフッタを出力しません。
-w	行の長さを指定します。デフォルトでは-s オプションが指定されないときには72桁、-s オプションが指定されているときには512桁となります。

■サンプルオペレーション

・1ページ80桁、25行で出力します。

```
$ pr -w80 -l25 file
```

・1ページにつき3列で表示します。

```
$ pr -3 file
```

・5ページ目から出力します。

```
$ pr +5 file
```

5.3.3 jgawk（パターン走査および処理用言語）

■書式

jgawk [POSIX or GNU style options] -f program-file [--] file ...

jgawk [POSIX or GNU style options] [--] program-text file ...

■機能

ファイルまたは標準入力からテキストファイルを読み込み、特定のパターンにマッチすると、指定した動作をします。言語の詳細については、市販されている awk の入門書などを参照してください。

■オプション

-f progfile	AWK のプログラムファイルを指定します。
-F fs	fs で指定した文字を入力フィールドセパレータとします。
-v var=val	プログラムを実行する前に変数 var に val をセットします。

■サンプルオペレーション

・file2 に対して file1.awk のスクリプトを実行します。

```
$ jgawk -f file1.awk file2
```

5.3.4　sed(ストリームエディタ)

■書式

sed [-an] command [file ...]

sed [-an] [-e command] [-f command_file] [file ...]

■機能

指定されたファイルまたは標準入力に対して編集セッションを実行し、標準出力に出力します。

■オプション

-e command	sed の編集コマンドを指定します。
-f command_file	sed の編集コマンドを並べたスクリプトを指定します。
-n	編集コマンドが実行された行のみ出力します。

■サンプルオペレーション

・file2 に対して file1 のスクリプトを実行します。

```
$ sed -f file1 < file2
```

・Zeus を Jupiter に置換して出力します。

```
$ sed -e 's/Zeus/Jupiter/g' < file
```

5.3.5 tail(ファイルの末尾の一部分の表示)

■書式

tail [-f | -r] [-b number | -c number | -n number] [file ...]

■機能

ファイルまたは標準入力から読み込み、入力の最後の部分を標準出力へ出力します。

■オプション

-f	ファイルの終わりに達しても終了せず、更なる入力を待ちます。
-r	出力する行を入力の逆順で出力します。
-b number	512 バイト単位で計算します。
-c number	バイト単位で計算します。
-n number	行数単位で計算します。

■サンプルオペレーション

・ファイルの最後の 10 行を出力します。

```
$ tail -n 10 file
```

・先頭から 10 行目以降を出力します。

```
$ tail +10 file
```

5.3.6　head(ファイルの最初の一部分の表示)

■書式

head [-n count] [file ...]

■機能

ファイルまたは標準入力の最初の数行を標準出力へ出力します。

■オプション

-n count	表示する行数を指定します。

■サンプルオペレーション

・ファイルの先頭から3行のみを出力します。

```
$ head -3 file
```

5.3.7 sort（テキストファイルのソート）

■書式

sort [-cmus] [-t separator] [-o output-file] [-T tempdir] [-bdfiMnr] [+POS1 [-POS2]] [-k POS1[,POS2]] [file...]

■機能

sort には、ソート、マージ、ソートの照合の 3 つの機能があります。いずれの機能もファイルまたは標準入力から読み込み標準出力へ出力します。

■オプション

-c	すでにソート済みかどうかをチェックします。
-m	すでにソートされたファイルをマージします。
-b	キーの前についている空白を無視します。
-f	小文字を大文字と見なします。
-i	非 ASCII 文字を無視します。
-M	月の名前の三文字の略称を比較します。(例 JAN < FEB< ... < DEC)
-n	数字を ASCII 順ではなく数値として比較します。
-r	逆順にソートします。
-o output-file	標準出力のかわりに output-file に出力します。
-t separator	フィールドのセパレータを指定します。
-u	-m オプションのときには同じ行がそれぞれある場合には一つのみ出力します。-c オプションのときには同じ行が存在するかどうかを確認します。
+POS1 [-POS2]	キーとして使うフィールドを指定します。ただし、フィールド番号は 0 から始まります。
-k POS1[,POS2]	キーとして使うフィールドを指定します。ただし、フィールド番号は 1 から始まります。

■サンプルオペレーション

・キーを数値としてソートします。

```
$ sort -n file
```

・フィールド 2 とフィールド 3 をキーとしてソートします。

```
$ sort +2 -4 file
```

・フィールドセパレータを:としてソートします。

```
$ sort -t:
```

・3番目から5番目までの文字をキーとしてソートします。

```
$ sort +.3 -.6
```

・file1とfile2を数値としてマージします。

```
$ sort -mn file1 file2
```

5.3.8 fold(長い行を指定した行幅へ折りたたむ)

■書式

fold [-w width] file ...

■機能

ファイルまたは標準入力から読み込み、指定された幅で改行文字を挿入して行を折たたみます。

■オプション

-w width 出力行の長さを指定します。

■サンプルオペレーション

・デフォルトでは 1 行を 80 文字に畳み込みます。

```
$ fold file
```

・file を 72 文字に畳み込みます。

```
$ fold -w72 file
```

5.3.9 cut(各々の行の一部分の切り出し)

■書式

cut -c list file ...

cut -f list [-d string] [-s] file ...

■機能

ファイルまたは標準入力から読み込み、各々の行から指定された部分を切り出します。

■オプション

-c list	文字位置によって指定します。
-d string	フィールドのデリミタとしてstringを用います。デフォルトのデリミタはタブです。
-f list	フィールドによって指定します。
-s	デリミタのない行を出力しません。

■サンプルオペレーション

・各行のフィールド2および3のみを出力します。

```
$ cut -f2,3 file
```

・文字桁位置4以外の桁を出力します。

```
$ cut -c1-3,5- file
```

5.3.10 paste(複数ファイルの並行マージ)

■書式

paste [-s] [-d list] file ...

■機能

複数のファイルを並行にマージします。つまり各入力ファイルは、出力では一つの列となります。

■オプション

-d list	列を区切るデリミタを指定します。
-s	複数のファイルからではなく一つのファイルから行を読み込んでマージします。つまり、i 番目のファイルは i 行目に出力されます。

■サンプルオペレーション

・1 列目に file1、2 列目に file2 を出力します。

```
$ paste file1 file2
<入出力例>
==> file1 <==
(1,1)
(2,1)
(3,1)

==> file2 <==
(1,2)
(2,2)
(3,2)

==> 出力 <==
(1,1)   (1,2)
(2,1)   (2,2)
(3,1)   (3,2)
```

・file1 の各行をマージしたものを 1 行目に、file2 の各行をマージしたものを 2 行目に表示します。

```
$ paste -s file1 file2
<入出力例>
(1,1)   (2,1)   (3,1)
(1,2)   (2,2)   (3,2)
```

5.3.11 uniq（隣接している重複行の検出または削除）

■書式

uniq [-c | -d | -u] [-f fields] [-s chars] [input_file [output_file]]

■機能

ファイルまたは標準入力から読み込み、隣接している重複行があると削除します。重複行は隣接していなければ発見されません。

■オプション

-c	各々の出力行の先頭に重複回数を表示します。
-d	重複しなかった行を出力しません。
-f fields	行頭からから fields 個のフィールドを無視しして比較します。
-s chars	行頭から chars 個の文字を無視しして比較します。
-u	重複していた行を出力しません。

■サンプルオペレーション

・重複している行を削除します。

```
$ sort -m file1 file2 | uniq
```

・重複回数を表示します。

```
$ uniq -c file1
```

5.3.12 wc（単語数、行数、バイト数のカウント）

■書式

wc [-clw] [file ...]

■機能

ファイルまたは標準入力から読み込み、行数、単語数、バイト数を計算して標準出力に出力します。

■オプション

-c	バイト数を計算します。
-l	行数を計算します。
-w	単語数を計算します。

■サンプルオペレーション

・file1 と file2 とその合計の場合について行数、単語数、バイト数を出力します。

```
$ wc file1 file2
```

・単語数のみを出力します。

```
$ wc -w file1
```

5.3.13　expand、unexpand（タブからスペースへの変換、そしてその逆）

■書式

expand [-tabstop] [-tab1,tab2,...,tabn] file ...

unexpand [-a] file ...

■機能

expandはファイルまたは標準入力から読み込み、タブをスペースに変換します。unexpandはファイルまたは標準入力から読み込み、スペースをタブに変換します。

■オプション

-tabstop	タブの桁数を設定。
-tab1,tab2,...,tabn	タブストップの位置を指定します。
-a	(unexpandのみ) デフォルトでは行頭のスペースのみをタブに変換しますが、このオプションを指定すると行頭のスペースだけでなくすべてのスペースの並びを強制的にタブに置き換えます。

■サンプルオペレーション

・2文字ごとにタブストップを設定して、タブ文字をスペースに展開します。

```
$ expand -2  file1
```

5.3.14 grep、egrep、fgrep（パターンにマッチする行の出力）

■書式

grep [-[[AB]]num] [-[CEFGVBchilnsvwx]] [-e] pattern | -ffile] [files...]

■機能

ファイルまたは標準入力から読み込み、パターンにマッチする行を検索します。egrep は grep -e と、fgrep は grep -F と同じです。

■オプション

grep のオプションには以下の 3 つがあります。

-G	pattern を基本的な正規表現として解釈します。
-E	pattern を拡張的な正規表現として解釈します。
-F	pattern を固定文字列として解釈します。

3 つの grep に共通するオプションは以下のとおりです。

-num	マッチした行の前後 num 行を出力します。
-A num	マッチした行の後 num 行を出力します。
-B num	マッチした行の前 num 行を出力します。
-C	-2 と同様です。
-V	バージョン番号を標準エラー出力に出力します。
-b	マッチした行の先頭からのバイト数を出力します。
-c	マッチした行数だけを出力します。
-e pattern	pattern をパターンとして使います。
-f file	検索パターンを file から得ます。
-h	複数のファイルを指定したときに行頭にファイル名が表示されるのを抑止します。
-i	大文字小文字の区別を無くします。
-L	出力する行が存在しなかったファイルはファイル名のみを出力するようにします。
-l	出力する行が存在するファイルはファイル名のみを出力するようにします。
-n	出力の各々の行に入力での行番号を出力します。
-q	通常の出力を抑止します。
-s	エラーメッセージを表示しません。
-v	一致しない行のみを出力します。
-w	単語として完全にマッチした行のみ出力します。
-x	完全に一致した行のみ出力します。

■サンプルオペレーション

・数字を含む行を表示します。

```
$ grep '[0-9] file
```

・"-"を検索文字に指定します。

```
$ grep -e- file
```

5.4 ページャ（テキストファイルの表示）

5.4.1 less（バックスクロール可能なページャ）

■書式

less [-[+]aBcCdeEfHimMnNqQrsSuUw] [-b bufs] [-h lines] [-j line] [-k keyfile] [-{oO} logfile] [-p pattern] [-P prompt] [-t tag] [-T tagfile] [-x tab] [-y lines] [-[z] lines] [+[+]cmd] [filename]...

■機能

less は more と同様のページャですが、バックスクロールが可能です。また大きなファイルであっても、ファイルのすべてを読み込まないのでより高速に動作します。

■オプション

-?	ヘルプを表示します。
-c	最も上の行から表示を始めます。
-C	-c オプションと同じですが、表示する前に画面をクリアします。
-hn	後方スクロールの最大行数を設定します。
-i	検索するときに大文字小文字を区別しません。
-m	冗長なプロンプトに設定します。
-M	-m よりもさらに冗長なプロンプトに設定します。
-n	行番号を表示しません。
-N	行の先頭に行番号を表示します。
-r	コントロール文字を見えるようにします。
-S	画面の幅よりも長くてはみ出る行ははみ出た部分を捨てて表示します。

■コマンド

・h,H

ヘルプを表示します。

・ SPACE 、^V、f、^F

1 画面進みます。

・z

1 画面進みます。

・RETURN、^N、^E、j、^J

1 行進みます。

・d、^D

半画面進みます。

・b、^B、 ESC v

1画面戻ります。

・w

1画面戻ります。

・y、^Y、^P、k、^K

1行戻ります。

・u,^U

半画面戻ります。

・r、^R、^L

画面全体を再表示します。

・R

画面を再表示します。バッファ内容は捨てられます。

・F

前に進みますが、ファイルの最後に達してもさらに読み続けようとします。

・g、<、 ESC <

ファイルの最初の行に行きます。

・G、>、 ESC >

ファイルの最後の行に行きます。

・p、%

ファイルの最初に行きます。

・/pattern

patternを前方に検索します。

・?pattern

patternを後方に検索します。

・n

後方検索を繰り返します。

・N

前方検索を繰り返します。

・q、:q、:Q、ZZ

lessを終了します。

・! shell-command

サブシェルで shell-command を実行します。

■サンプルオペレーション

・mule2.tar.gz の内容を less で表示します。

```
$ tar tvzf mule2.tar.gz | less
```

5.5　漢字コードの変換

5.5.1　kc（漢字コード変換プログラム）

■書式

kc [-k][-bXX][-f][-v][-[input kanjicode][output kanjicode]] [file(s)]

■機能

漢字コードを変換、識別をします。

■オプション

-k	漢字コードの識別のみします。
-bXX[k]	漢字コードを識別するのに調べる入力文字列のサイズを指定します。
-f	バッファリング出力をしません。
-v	冗長モード。

・漢字コードの指定

s	shift jis
e	euc
j	jis { ESC$B [kanji'83], ESC(B [ascii], ESC(I [kana(7bit)] }
J	jis { ESC$B [kanji'83], ESC(J [roman] }
o	oldjis { ESC$@ [kanji'78], ESC(J [ascii], ESC(I [kana(7bit)] }
c	ctext { ESC$(B [kanji'83], ESC(B [ascii], ESC)I [kana(8bit)] }

■サンプルオペレーション

・漢字コードを調べます。

```
$ kc -k file1
SJIS
```

・漢字コードを SJIS から JIS に変換します。

```
$ kc -sj file1
```

5.6　アーカイバと圧縮ツール

5.6.1　tar（テープアーカイバ）

■書式

tar [[-]] [bundled-options] [[gnu-style-flags]] [tarfile] [blocksize] [exclude-file] [filenames] [-C directory-name]

■機能

■コマンド

tarfile と呼ばれる形式のアーカイブファイルを扱うためのプログラムです。以下の機能のうちの１つを選択します。

-A	アーカイブにファイルを追加します。
-c	新しいアーカイブを作ります。
-d	アーカイブとファイルシステムの違いを調べます。
--delete	アーカイブからファイルを削除します。
-r	アーカイブの最後にファイルを追加します。
-t	アーカイブの内容を表示します。
-u	アーカイバの中のファイルよりも新しいファイルのみ追加します。
-x	アーカイブからファイルを取り出します。

■オプション

-b N	ブロックサイズを N×512 バイトに設定します（デフォルトでは N＝20 です）。
-B	短いブロックを完全なブロックになるように再構成します。
-C DIR	展開するディレクトリを DIR に変更します。
-f name	アーカイブファイル名またはデバイスを name に設定します。
-h	シンボリックリンクのときにはシンボリックリンクとしてではなく、参照先のファイルを書き込みます。
-k	すでにファイルが存在しているときには上書きしません。
-N DATE	DATE よりも新しいファイルのみ保存します。
-O	ファイルを標準出力に出力します。
-v	冗長モード。処理の途中経過を表示します。
-Z	アーカイブを compress に通して圧縮または展開します。
-z	アーカイブを gzip に通して圧縮または展開します。

■サンプルオペレーション

・mh-6.8.3-JP.2c.tar.gzの中身を冗長モードで表示します。

```
$ tar tvzf mh-6.8.3-JP.2c.tar.gz
```

・less-290.tar.gzを展開します。

```
$ tar xvzf less-290.tar.gz
```

・ホームディレクトリ以下をアーカイブに追加します。

```
$ tar cvf file.tar ~
```

5.6.2 unzip（zip形式アーカイブファイルの展開）

■書式

unzip [-Z] [-opts[modifiers]] file[.zip] [list] [-x xlist] [-d exdir]

■機能

zip ファイルを展開します。

■オプション

-p	ファイルを展開しパイプへ出力します。
-f	アーカイブを新たにします。
-u	アーカイブを更新します。
-x xlsit	xlist に含まれるファイルについては対象外とします。
-l	アーカイブ内のファイルをリストします。
-t	アーカイブのデータをチェックします。
-z	アーカイブのコメントを表示します。
-d exdir	exdir へファイルを取り出します。
-q	静粛モード。-qq とするとさらに静粛になります。
-n	ファイルがすでに存在するとき上書きしません。
-a	テキストファイルを自動変換します。
-o	確認のプロンプトを出さずに上書きします。
-aa	すべてのファイルをテキストファイルとして扱います。
-j	アーカイブ内のディレクトリを無視します。
-v	冗長モード。
-C	大文字小文字を無視します。
-L	ファイル名を小文字にします。

■サンプルオペレーション

・file2 以外のすべてのファイルを file1.zip から取り出します。

```
$ unzip file1 -x file2
```

・file.zip の内容を標準出力に出力します。

```
$ unzip -p file
```

5.6.3　uuencode、uudecode（バイナリファイルのテキストエンコードとデコード）

■書式

uuencode [file] name

uudecode [file ...]

■機能

uuencode と uudecode は ASCII データしかサポートしない伝送媒体でバイナリファイルを転送するために用いられます。uuencode は指定されたファイルを読み込んで印字可能文字だけでコード化し標準出力に出力します。uudecode は uuencde されたファイルを元のバイナリファイルにデコードします。

■オプション

name　　uudecode されたときのファイル名を指定します。

■サンプルオペレーション

・file1 をコード化して file3 に書き出します。そして、デコードしたときのファイル名が file2 になるようにします。

```
$ uuencode file1 file2  > file3
```

・file3 をデコードします。

```
$ uudecode file3
```

5.6.4 lha(高圧縮書庫管理プログラム)

■書式

lha [-]{axelvudmcp}[qvnfodizg012][w=<dir>] archive_file [file...]

■機能

アーカイバ機能を持ったファイル圧縮プログラムです*1。

■コマンド

a	凍結して書庫に加えます。
x、e	書庫内からファイルを解凍します。
l、v	書庫内のファイルを一覧表示します。
u	書庫内のファイルを更新します。
d	書庫内のファイルを削除します。
m	書庫内のファイルを移動します。
c	新しい書庫を作成します。
p	書庫内のファイルを標準出力に出力します。
t	書庫内のファイルを検査します。

■オプション

q	静粛モード。
v	冗長モード。
n	対象となるファイル名を表示するのみで、実際に凍結はしません。
f	解凍するときに上書きします。
t	ファイルをテキストファイルと見なします。
o	LHarc でも解凍できる書庫を作成します。
w=<dir>	解凍するディレクトリを指定します。
d	凍結した後、元のファイルを削除します。
i	ディレクトリのパスを無視します。
z	ファイルを圧縮しません。
g	一般的なフォーマットを使います。
0/1/2	ヘッダーレベルを指定します。
e	テキストを EUC から、または EUC へ変換します。

■サンプルオペレーション

・/home 以下のサブディレクトリーを再帰的に書庫/tmp/home.lzh に加えます。

```
$ lha a /tmp/home.lzh /home
```

---

*1 lha は、吉崎栄泰氏がオリジナルの MS-DOS バージョンを作成し、沖勝氏が UNIX に移植したフリーソフトウェアです。LHA for MS-DOS Copyright © 1988-1992 H.Yoshizaki、LHA for UNIX Copyright © 1992 Masaru Oki。

5.6.5 compress、uncompress（データの圧縮と展開）

■書式

compress [-cfv] [-b bits] [file ...]

uncompress [-cfv] [file ...]

■機能

compress は adaptive Lempel-Ziv コードを使ってファイルを圧縮します。圧縮されたファイルには.Z の拡張子が付加されます。uncompress は compress で圧縮されたファイルを元に戻します。

■オプション

-b bits	共通部分の文字列コードの最大コード長を指定します。
-c	出力を標準出力に出力します。
-f	圧縮してもサイズが小さくならなくても圧縮を強行します。
-v	各々のファイルについて圧縮度のパーセンテージを表示します。

■サンプルオペレーション

・file1 を圧縮します。

```
$ compress file1
```

・file1.Z を元に戻します。

```
$ uncompress file1.Z
```

5.6.6 gzip、gunzip、zcat(データの圧縮と展開)

■書式

gzip [-acdfhlLnNrtvV19] [-S suffix] [name ...]

gunzip [-acfhlLnNrtvV] [-S suffix] [name ...]

zcat [-fhLV] [name ...]

■機能

gzipはLempel-Ziv法(LZ77)を用いてファイルを圧縮します。圧縮されたファイルには.gzの拡張子が付加されます。gunzipはgzipで圧縮されたファイルを元に戻します。zcatはgzipされたファイルまたは標準入力から読み込んで元に戻したデータを標準出力に出力します。

■オプション

-a	圧縮を解除するときテキストファイルの行末コードを環境によって自動的に変えます。
-c	出力を標準出力に出力します。
-d	圧縮を解除します。
-f	強制的に実行します。
-h	ヘルプを表示します。
-l	圧縮されたファイルに対しての情報を表示します。
-n	圧縮するときに元のファイル名とタイムスタンプをセーブしません。
-N	圧縮するときに元のファイル名とタイムスタンプをセーブします。
-q	すべての警告を抑止します。
-r	ディレクトリーを再帰的に圧縮します。
-S .suf	拡張子を.gzのかわりに.sufを使用します。
-t	圧縮されたファイルの整合性をテストします。
-v	圧縮時や非圧縮時にファイル名や圧縮率など冗長な情報を表示します。
-数字	圧縮の程度を調整します。-1のときはもっとも高速に圧縮されます(圧縮率は低くなります)。-9のときはもっとも速度が遅く圧縮率は高くなります。デフォルトは-6です。

■サンプルオペレーション

・圧縮率が最も高くなるように圧縮します。

```
$ gzip -9 file
```

・圧縮されたファイルを元に戻します。

```
$ gunzip file.gz
```

・圧縮されたファイルをパイプに通します。

```
$ zcat file.gz | tar xvf -
```

5.7　ネットワーク関連

5.7.1　ftp（ARPANETのファイル転送プログラム）

■書式

ftp [-v] [-d] [-i] [-n] [-g] [host]

■機能

File Transfer Protocol で遠隔のネットワークとファイル転送します。

■オプション

-v	remote server からのすべての応答を表示します。
-n	初期接続時に”自動ログイン”を試みます。
-i	複数のファイルを転送しているときに対話モードのプロンプトを抑止します。
-d	デバッグモードにします。

■コマンド

・! [command [args]]

ローカルホストで対話モードのシェルを立ちあげます。command を指定するとそのコマンドが直接実行されます。

・ascii

ASCII モードで転送します（テキストファイルを転送する場合に使用）。

・binary

BINARY モードで転送します（バイナリファイルを転送する場合にしよう）。

・bye

FTP セッションを終了し、FTP コマンドを終了します。

・cd remote-directory

リモートマシンのカレントディレクトリを変更します。

・close

FTP セッションを終了し、コマンドインタプリタに戻ります。

・delete remote-file

リモートマシン上のファイルを削除します。

・dir [remote-directory] [local-file]

指定されたリモートディレクトリの内容を表示します。

・get remote-file [local-file]

リモートマシン上の remote-file をローカルマシンに転送します。

・help

ヘルプを表示します。

・lcd [directory]

ローカルマシン上のカレントディレクトリを変更します。directory を指定しなかったときにはユーザのホームディレクトリになります。

・ls [remote-directory] [local-file]

リモートマシン上の指定されたディレクトリの内容を出力します。

・mget remote-files

複数個のファイルを get します。

・mkdir directory-name

リモートマシン上にディレクトリを作ります。

・mput lcoal-files

複数個のファイルを put します。

・open host [port]

指定された FTP サーバと接続を確立します。

・prompt

対話モードのプロンプトをオン／オフします。

・put local-file [remote-file]

指定されたローカルファイルをリモートマシン上に格納します。

・pwd

リモートマシン上のカレントディレクトリを表示します。

・quit

bye に同じです。

・rename [from] [to]

リモートマシン上のファイル名を変更します。

・rmdir directory-name

リモートマシン上のディレクトリを削除します。

■サンプルオペレーション

・リモートマシン sapc00 にログインして、リモートマシンの/tmp/nkf1.4-tar.gz という
　バイナリファイルを、ローカルマシンの/tmp に転送します。

```
$ ftp
ftp> open sapc00
Connected to sapc00.
220 sapc00 FTP server (Version 6.00) ready.
Name (sapc00:bow-user): bow-user
331 Password required for bow-user.
Password: ********
230 User bow-user logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd /tmp
250 CWD command successful.
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for '/bin/ls'.
total 18
-rw-r--r--  1 bow-user  bin  7582 Feb 10 18:11 nkf-1.4.tar.gz
drwxrwxrwt  2 bin       bin   512 Feb 10 03:07 vi.recover
226 Transfer complete.
ftp> lcd /tmp
Local directory now /tmp
ftp> get nkf-1.4.tar.gz
local: nkf-1.4.tar.gz remote: nkf-1.4.tar.gz
200 PORT command successful.
150 Opening BINARY mode data connection for 'nkf-1.4.tar.gz' (7582 bytes).
226 Transfer complete.
7582 bytes received in 0.0067 seconds (1.1e+03 Kbytes/s)
ftp> bye
221 Goodbye.
```

5.7.2 telnet（TELNETプロトコルのユーザーインターフェイス）

■書式

telnet [-8EFKLacdfrx] [-S tos] [-X authtype] [-e escapechar] [-k realm] [-l user] [-n tracefile] [host [port]]

■機能

TELNET プロトコルを用いてほかのマシンと通信します。

■オプション

host	接続先の正式なホスト名、エイリアス（別名）、IP アドレスなどを指定します。
port	接続するポート番号を指定します。デフォルトは telnet port です。

■コマンド

telnet では、通信中にエスケープ文字（デフォルトでは Ctrl ＋]）を入力することで、コマンドモードに移行することができます。

・close

TELNET セッションを終了しコマンドラインに戻ります。

・open　host [[-l] user][-port]

host という名前のホストと接続します。ポート番号を指定しない場合にはデフォルトポートの TELNET に接続を試みます。

・quit

TELNET セッションを終了し、telnet コマンドを終了します。

・z

telnet コマンドをサスペンドします。

・! [command]

対話モードのシェルを立ちあげます。command を指定するとそのコマンドが直接実行されます。

・? [command]

ヘルプを表示します。

■サンプルオペレーション

・マシン名が sapc00 のマシンに接続します。

```
$ telnet
telnet> open sapc00
Trying 133.152.8.1...
Connected to sapc00.
Escape character is '^]'.

   FreeBSD (sapc00) (ttyp2)

login: bow-user
Password: ********
Last login: Sat Feb 10 18:17:25 from sapc00
Copyright (c) 1980, 1983, 1986, 1988, 1990, 1991, 1993, 1994
        The Regents of the University of California.   All rights reserved.

FreeBSD 2.0.5-RELEASE (SAPC00) #0: Fri Nov 24 19:16:17 JST 1995

Welcome to FreeBSD!

bow-user@sapc00[101]
```

・ここで telnet コマンドのエスケープキャラクタである^] を入力すると telnet のコマンドプロンプトに戻ります。

```
telnet> quit
Connection closed.
```

・smtp のポート（ポート番号 25）に接続します。

```
$ telnet sapc00 25
```

5.8 その他

5.8.1 file（ファイルタイプの推測）

■書式

file [-c] [-z] [-L] [-f namefile] [-m magicfile] file ...

■機能

ファイルを読み出して、そのファイルの種類を推測します。

■オプション

-z	圧縮されたファイルの中身を調べることを試みます。
-c	マジック番号ファイルの中身をチェックします。
-f namefile	ファイルタイプを識別するためのファイルを列挙したファイル namefile を指定します。
-L	シンボリックリンクのときには、シンボリックリンク自体ではなく参照されているファイルについて調べます。

■サンプルオペレーション

```
$ file /bin/hostname
/bin/hostname: demand paged executable
```

5.8.2　find（サブディレクトリ中のファイルを検索）

■書式

find [-H | -L | -P] [-Xdx] [-f file] [file ...] expression

■機能

ディレクトリツリーを下っていきそのツリーの中にあるすべてのファイルひとつづつに対して、expressionを評価します。expressionが真の場合には、処理が行なわれます。

■オプション

-H	コマンドラインで指定されたシンボリックリンクに対しては参照されているファイルについて実行しますが、サブディレクトリ以下のシンボリックリンクに関してはシンボリックリンク自身について実行します。
-L	シンボリックリンクのときには参照されているファイルについて常に実行します。
-P	シンボリックリンクのときにはシンボリックリンク自身について実行します。
-X	xargs(1)と連携して安全に検索します。
-d	ディレクトリよりもエントリを先に検索します。
-f	
-x	デバイス番号の異なるディレクトリーには下っていきません。

■演算子

-atime n	ファイルがn日以内にアクセスされた場合には真になります。
-ctime	ファイルがn日以内に変更された場合には真になります。
-exec utility [argument ...];	指定したutilityを実行しその終了ステータスがゼロの場合には真になります。
-fstype type	ファイルの属するファイルシステムがtype出会った場合には真になります。
-group gname	ファイルのgnameに属している場合には真になります。
-inum n	ファイルのiノード番号がnの場合には真になります。
-links n	ファイルがn個のリンクを持つ場合には真になります。
-ls	常に真となります。ファイルについての情報を標準出力に出力します。
-mtime n	ファイルがn日以内に修正された場合に真になります。
-ok utility [argument ..];	ユーザに確認を要求する点をのぞいて-execオプションと同じです。
-name pattern	名前がpatternに一致するときには真になります。
-newer file	fileよりもファイルが新しいときには真になります。
-nouser	ファイルが知らないユーザのものであれば真になります。
-nogroup	ファイルが知らないグループのものであれば真になります。
-path pattern	パス名がpatternに一致するときには真になります。
-perm [-mode]	ファイルのアクセスモードが一致すれば真になります。
-print	常に真となります。現在のファイルのパス名を標準出力に出力します。

-print0	常に真となります。現在のファイルのパス名を標準出力に出力し、最後に NULL 文字を出力します。
-prune	常に真となります。現在のファイルで検索を停止します。
-size n[c]	ファイルの容量が一致したときには真になります。
-type t	ファイルのタイプが一致したときには真になります。
-user uname	ファイルが uname に属している場合には真になります。

■サンプルオペレーション

- ルートディレクトリ以下から find.1 という名前のファイルを探し、見つかったら標準出力に出力します。

```
$ find / -name find.1 -print
```

- ホームディレクトリ以下の、"~"で終わるすべてのファイルを削除します（csh から実行する場合）。

```
$ find ~ -name "*~" -exec rm {} \;
```

5.8.3 date(日付と時間の表示)

■書式

date [-nu]

■機能

引数がない場合には現在の日付と時間を単に表示します。BOW上では、日時の変更はできません。

■オプション

-u　　世界標準時で表示、設定します。

■サンプルオペレーション

・現在の日付、時間を表示します。

```
$ date
Sun Mar 31 23:07:34 JST 1996
```

5.8.4　cal（カレンダーの表示）

■書式

cal [-jy] [month [year]]

■機能

簡単なカレンダーを表示します。

■オプション

-j　　ユリウス暦のカレンダーを表示します。
-y　　現在の年のカレンダーを表示します。

■サンプルオペレーション

・西暦 2001 年の 1 月のカレンダーを表示します。

```
$ cal 1 2001
```

5.8.5　echo（文字列の表示）

■書式

echo [-n] [string ...]

■機能

引数を標準出力に書き込みます。

■オプション

-n　　改行文字を出力しません。

■サンプルオペレーション

・環境変数 PATH の値を表示します。

```
$ echo $PATH
```

5.8.6　printenv、env（環境変数の表示、設定）

■書式

printenv [name]

env [-] [name=value ...] [command]

■機能

printenv は環境変数名とその値を表示します。env は子プロセスに対して環境変数を設定し、実行します。

■オプション

-　　今までの環境を引き継がずに消去してから環境変数を設定します。

■サンプルオペレーション

・環境変数 MANPATH の値を表示します。

```
$ printenv $MANPATH
```

5.8.7　sleep（指定された時間の実行休止）

■書式

sleep seconds

■機能

seconds 秒間、プロセスの実行を休止します。

■オプション

secods　　実行休止する時間を秒単位で指定します。

■サンプルオペレーション

・1時間後に finger　@bow01 を実行します。

```
$ (sleep 3600;finger @bow01)&
```

6章

ネットワークの利用

BOW 1.5 では、ネットワーク機能が拡張されています。BOW から利用できるネットワークには、一般の BSD システムでも標準になっている「TCP/IP ネットワーク」と、Windows NT や Windows 95 などで構成される「Microsoft Network」の 2 種類があります。また、接続形態として Ethernet などによる LAN 接続と、電話回線を利用したダイヤルアップ接続の 2 種類があります。Windows 95 では、これらのネットワーク機能はすべてシステムに標準で備わっているものを利用できます。Windows 3.1 でネットワーク機能を使う場合は、使用するネットワーク環境に応じて、別途ネットワークソフトウェアを購入する必要があります。これらのネットワーク製品の種類によっては、一部の機能が使えなかったり、動作が不安定になる恐れがありますので、BOW 1.5 でネットワーク機能を利用する場合は、Windows 95 での利用を推奨します。

6.1 TCP/IPネットワークの利用について

Windows 95 上で BOW1.5 を使うと、Windows に付属する TCP/IP ネットワーク機能を利用して、他の UNIX ホストなどとネットワークアプリケーションを利用した通信ができるようになります。ネットワークの利用形態は、お使いの環境によって大きく異なるため、本マニュアルでは最低限の設定についてのみ解説します。お使いのネットワーク環境での利用については、ネットワークアプリケーションのマニュアルを参照したり、ネットワーク管理者に問い合わせてください。

6.1.1 Windows 95の設定

Windows 95 でのネットワークの設定は、コントロールパネルの［ネットワーク］アイコンで行ないます。ここでは、すでに UNIX ホストなどを含む TCP/IP ネットワークに接続されていることを前提に解説します。すでに Windows 95 上で TCP/IP による通信の設定が行なわれている場合は、この節は読む必要はありません。「BOW 側の設定」に進んでください。

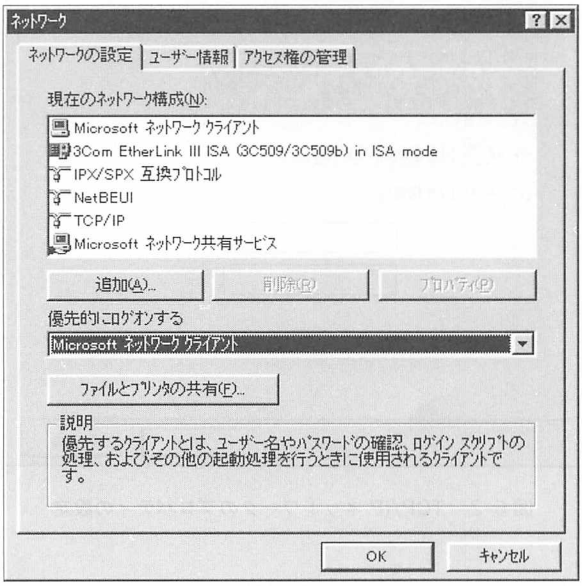


図 6-1 ［ネットワークの設定］ダイアログボックス

まず、コントロールパネルの[ネットワーク]アイコンをダブルクリックし、表示されたダイアログボックスで[ネットワークの設定]を選択してください(図6-1)。ここで、[現在のネットワーク構成(N)]のリスト中に、「TCP/IP -> [お使いのネットワークカード名]」という項目がすでに存在している場合は手順2に進んでください。この項目がない場合は、[追加(A)]ボタンをクリックします。[ネットワーク構成ファイルの追加]というダイアログが表示されるので、[プロトコル]の項目を選択して、[追加(A)]のボタンをクリックします。次に表示される[ネットワークプロトコルの選択]というダイアログでは、[製造元(M)]で「Microsoft」を、[ネットワークプロトコル]で「TCP/IP」を選択します。選択した後に[OK]ボタンをクリックすると、先ほどの[ネットワーク]ダイアログに戻ります。リスト中に「TCP/IP -> [ネットワークカード名]」の項目が追加されたはずです。

次に、さきほど追加された「TCP/IP -> [ネットワークカード名]」の項目をマウスで選択し、[プロパティ]のボタンをクリックします。すると図6-2のようなダイアログボックスが表示されます。ここには、お使いのネットワーク環境に応じて適切な値を入力する必要があります。ここでの設定方法については、ネットワーク環境によってまったく異なりますので、ネットワーク管理者に問い合わせて正しい設定を行なってください。

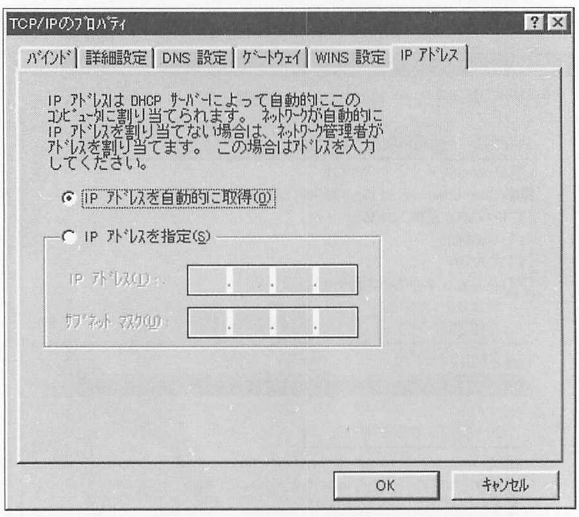


図6-2　TCP/IP ネットワークのプロパティの設定

すべての設定が終わると、システムを再起動する旨のダイアログボックスが表示されるので、指示にしたがってシステムを再起動します。システムが正常に立ち上がり、Windows 95からTCP/IPによる通信が正常に行なえる(C:\WINDOWS\PING.EXE、

C:\WINDOWS\TELNET.EXE などが正常に使える）ことを確認したら、BOW 側の設定に進みます。

6.1.2 BOW側での設定

BOW 側で行なう設定は、一般の BSD システムでの TCP/IP アプリケーションの設定とほぼ同じですが、以下のような設定については必要ありません。

- ifconfig などによるネットワークインターフェイスの設定
- route などによるネットワークアドレスやルーティングテーブルの設定
- inetd.conf などによるネットワークデーモンの設定

BOW でおもに必要になる設定は以下のとおりです。

- /etc/hosts の設定、または、DNS(bind)の設定(/etc/host.conf、/etc/resolv.conf)
- /etc/services、/etc/protocols の設定

このうち、/etc/services、/etc/protocols については、デフォルトのまま変更する必要はありません。これらは、特殊なネットワークアプリケーションを実行する際に変更しますが、そのようなアプリケーションでは、かならずドキュメントに変更方法が記載されていますので、その指示に従ってください。ここでは、/etc/hosts、/etc/host.conf、/etc/resolv.conf の設定についてだけ解説します。これらのファイルの設定は、DNS を使う場合と使わない場合で異なりますので、それぞれの場合に分けて解説します。

■ DNS を使わない場合

DNS を使わない場合、接続先のマシンのホスト名とアドレスの対応表を、ローカルの環境に作成する必要があります。この対応表が/etc/hosts です。接続する必要のあるすべてのホストについて、次のような形式で/etc/hosts に記述します。

```
127.0.0.1          localhost
XXX.XXX.XXX.XXX    hostname alias-name
...
```

ここで、XXX.XXX.XXX.XXX の部分には、IP アドレスを、hostname にはその IP アドレスを持つホスト名、alias-name にはそのホストの別名を記述します（alias-name は省略できます）。127.0.0.1 のアドレスをもつホスト（localhost）は、自分自身を表わす IP アドレスとして必要なものなので、削除しないでください。/etc/hosts には、このように

IP アドレスとホスト名のリストを、必要なだけ記述します。

次に/etc/host.conf ですが、このファイルは、次の1行だけ記述します。

```
hosts
```

DNS を使わない場合、/etc/resolv.conf を記述する必要はありません。

■DNS を使う場合

DNS とは、ネットワーク上にホスト名と IP アドレスの対応表をもつサーバー（ドメインネームサーバー）を置いて、ホスト名と IP アドレスのデータをそのサーバーですべて管理するシステムのことです。BOW から DNS を利用する場合、Windows 95 側で DNS の設定を行なう必要はかならずしもありませんが、問題が起きたときの切り分けのために、Windows 95 側でも DNS の設定を行なっておいたほうがよいでしょう。Windows 95 での DNS の設定については、ネットワーク管理者に問い合わせてください。ここでは、BOW 上での設定についてのみ解説します。

DNS で設定が最低限必要なファイルは、/etc/host.conf と/etc/resolv.conf の2つです。/etc/hosts については、DNS が正常に働かない場合に備えたり、DNS サーバーに登録されていないホストにアクセスする必要がある場合に、前項と同じように設定します。

/etc/host.conf の内容は、DNS のみ使用する場合は、

```
bind
```

の1行のみ記述します。DNS を利用し、DNS が失敗した場合には/etc/hosts の内容を参照する場合は、

```
bind
hosts
```

のように2行記述します。

/etc/resolv.conf の内容は以下の通りです。

```
domain   [your-domain]
nameserver   [your-name-server1]
nameserver   [your-name-server2]
...
```

ここで、[your-domain]の部分には、お使いの環境でのドメイン名、[your-name-server1]と[your-name-server2]には、ドメイン内でのネームサーバーの IP アドレスを記述しま

す。ネームサーバーが1つしかない場合には、nameserverの行は1行だけでかまいません。また、3つ以上ネームサーバーが利用できる場合には、3行以上同じように記述します。たとえば、「testdomain.or.jp」というドメインで、133.152.XXX.YYY、133.152.XXX.ZZZの2つのネームサーバーがある場合には、

```
domain       testdomain.or.jp
nameserver   133.152.XXX.YYY
nameserver   133.152.XXX.ZZZ
```

のように記述します（XXX、YYY、ZZZには実際には数値が入る）。

以上の設定を行なったら、いったんすべてのBOWプログラムを終了します。次にBOWのプログラムを起動するときから、ネットワークアプリケーションが利用できるようになります。

6.1.3 ダイヤルアップIP接続を使う場合

ダイヤルアップIP接続でネットワークアプリケーションを使う場合、基本的な設定事項はLANによる場合と変わりません。Windows側での設定については、Windowsのマニュアルを参照してください。BOW側では、接続先の環境に応じてDNSの設定を行ないます。BOW上のBSDアプリケーションからネットワークへアクセスが発生すると、自動的にダイヤルアップの処理（ダイヤルアップを行なうダイアログボックスが表示される）が実行されます。

6.2 Microsoftネットワークについて

Windows NTやWindows 95で標準的に使われているネットワーク環境が「Microsoftネットワーク」です。Microsoftネットワークでは、

■マシン間でのファイルやディスクドライブの共有
■プリンタやモデムの共有
■独自プロトコルでのメール転送
■アプリケーション間でのデータ共有

などのサービスが提供されます。これらのなかで、BOWから利用できるのは、ディスクドライブとプリンタの共有サービスです。

プリンタに関しては、BOW側から直接接続先を指定することはできませんが、コントロールパネルの[プリンタ]アイコンなどで「通常使うプリンタ」にリモートプリンタを指定すれば、そのプリンタに対して出力することが可能になります。

ファイルやドライブの共有に関しては、次のような方法で利用することができます。

- エクスプローラの[ツール(T)][ネットワークドライブの割り当て(M)]でドライブ名を割り当てたネットワークドライブに対して、/dfs/[ドライブ名]の形式でアクセスできる。
- BOW Configurationの[Network]ダイアログで指定したネットワークドライブに対しては、/dfs/[ホスト名]/[共有名]の形式でアクセスできる。

ネットワークドライブの設定方法については、本マニュアルの4章の「BOW Configuration」の項を参照してください。

7章
BOWの仕様

この章では BOW に対するソフトウェア、ハードウェアに関する制限やデバイスの実装について説明します。BOW と連携するプログラムや BSD プログラムを作成する際には、この章の内容を前提とする必要があります。

7.1　BOW 1.0から拡張された仕様

BOW 1.5 では、Windows 95 に対応したことで、ファイルシステムの仕様をはじめとしていくつかの拡張が行なわれています。これらの仕様拡張のうちのいくつかは、Windows 3.1 の環境でも有効ですが、一部の機能については動作が不安定になる場合もありますのでご注意ください。BOW 1.5 は、Windows 95 での使用を推奨しています。

7.1.1　ファイル名の8.3制限の緩和

Windows 95 では、従来の DOS や Windows で使われてきた FAT ファイルシステムを拡張した「VFAT」ファイルシステムが採用されています。VFAT ファイルシステムは FAT ファイルシステムと相互の互換性がありますが、ドライブ名とパス名を含めたファイル名の長さが、最大 259 文字までに拡張されています。また、使用できる文字種の制限も緩和され、BSD のファイルシステムで使える文字種のほとんどが VFAT でも使えるようになりました。このため、BOW 1.0 で行なっていたファイル名の変換については、BOW 1.5 を Windows 95 上で使う場合については原則として行なわないようになりました。VFAT のファイル名の仕様上の制限については、**表 7-1**を参照してください。

表 7-1　VFAT ファイルシステムのファイル名の制限事項

制限事項	制限値
ファイル名の最大長	255 文字まで
パス名の最大長	259 文字まで
ファイル名に含めることのできない文字	″ ＊ ／ ： ＜ ＝ ＞ ？ ￥ ｜

VFAT ファイルシステム上のファイルに、DOS または Windows アプリケーションからアクセスする場合、長い形式のファイル名と、8.3 形式に縮められた短い別名のどちらでも同じようにアクセスできるようになっていますが、BOW では、長い形式と短い形式の両方の名前があるファイルの場合、長いファイル名によるアクセスのみ許すようになっています。たとえば、VFAT 上で 123456789.ABCD というファイルは、123456~1.ABC という別名でもアクセスできますが、BOW 上の BSD プログラムからは 123456789.ABCD という名前でのみアクセスできます。123456~1.ABC というファイルを開こうとすると、エラーになります。

Windows 3.1上でBOW 1.5を使う場合は、171ページで述べるファイル名の制限とファイル名の変換規則が適用されます。

7.1.2　シンボリックリンクとハードリンクのエミュレーションの追加

BOW 1.5では、Windowsのレジストリデータベース機能を利用して、ファイルのシンボリックリンクとハードリンクのエミュレーションが可能になりました。これにより、従来のln(1)コマンドのエミュレーションは廃止されました。シンボリックリンクは、ディレクトリ上にリンク名のダミーのファイルを作成し、レジストリにそのリンク先の情報を保存します。ハードリンクの場合は、ファイルの実体を@nnnnnnn (nnnnnnnの部分は7桁の整数) という名前のファイルに変更して、すべてのリンクはこのファイルへのシンボリックリンクとして再構成します。レジストリには、この実体ファイルを参照しているリンクの数 (リンクカウンタ) を保存しておき、このファイルへのリンクが増減するたびにリンクカウンタの値も更新するようになっています。

BOW 1.5のリンクは、ファイルシステムの機能ではなく、Windowsのレジストリデータベースを応用しているものなので、BOW内部で作成したリンクファイルや実体ファイルを、Windowsの他のアプリケーションやファイルマネージャ、エクスプローラなどで削除したり名前を変えたりすると、リンク情報が破壊される場合がありますので注意してください。なお、リムーバブルメディア (フロッピーディスクやMOなど) 上に作成したリンクファイルについては、メディアのボリュームシリアル番号によって管理しているので、複数のディスク間でリンクが混乱することはないようになっています。

なお、Rockridge Extension形式のCD-ROMでは、従来からリンクファイルがサポートされていますが、BOW 1.5でもBOW 1.0と同様に、このリンクファイルをサポートしています。

7.1.3　ネットワークドライブへのアクセス機構の追加

4章の88ページで解説しているように、Microsoftネットワークの環境下では、BOW上のBSDプログラムから、Microsoftネットワークの共有ドライブにアクセスできるようになりました。なお、NetWareやPC NFSなどの他のNetwork OSの環境でも、ドライブ名を割り当てる方法を使えばアクセスは可能です。

7.1.4　/dev/waveデバイスの追加

.WAV 形式の音声ファイルを再生する/dev/wave デバイスが追加されました。cat file.WAV > /dev/audio などとすることで、直接音声ファイルの再生が可能です。再生できるオーディオファイルは WAV 形式のものに限ります。

7.1.5　/dev/consoleデバイスの追加

/dev/console に対して文字列を出力することで、デバッグ用のウィンドウに文字列を出力できるようになりました。BOWPS.EXE に対して行なわれた拡張です。この機能を使うときはあらかじめ BOWPS を起動してから、コントロールメニューから [Open Console] を選択しておく必要があります。

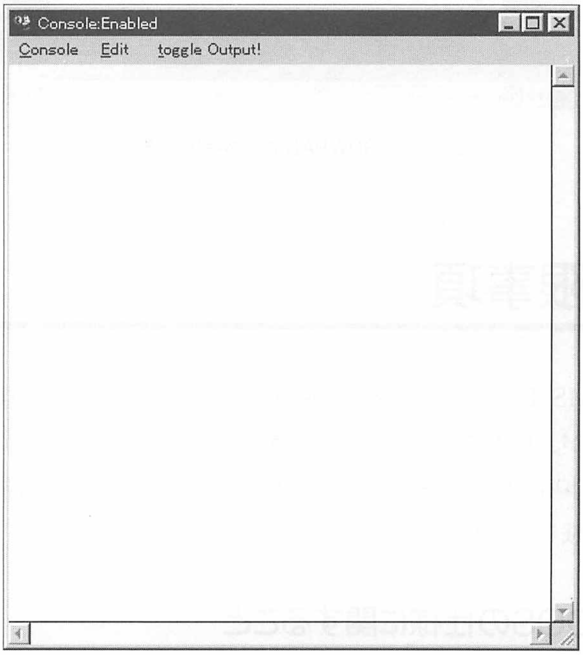


図 7-1　デバッグ用コンソールの表示 (BOWPS)

7.1.6　BOWPADに文字列検索機能を追加

BOWPADのコントロールメニューに[Search Text]が追加されました。これは、CANONICALモードのときのみ有効です。

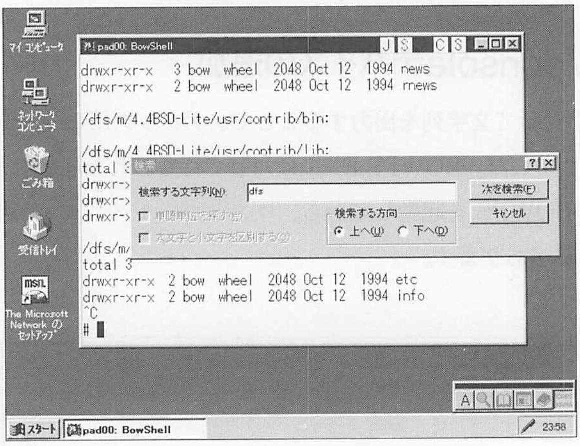


図7-2　BOWPADの文字列検索機能

7.2　制限事項

BSDの機能をMS-DOS、Windowsで動作させるためにさまざまな機能の変更や拡張が行われていますが、根本的に解決できない問題もありました。ここでは、それらの使用上の注意点や制限事項を解説します。なお、Windows 3.1で使う場合に固有の制限については、その旨を記載してあります。

7.2.1　MS-DOSの仕様に関すること

- オープン中のファイルに対しunlink(2)を発行してもすぐにはunlinkされません。unlinkはクローズ時に行われます。
- lseek(2)で未writeの領域を作成したとき0で埋められません（ufs holeは作ることができない）。
- ディレクトリに対しutime(2)（タイムスタンプの変更）はできません。

- ディレクトリの内容に変更があってもディレクトリのタイムスタンプは変わりません。ディレクトリのタイムスタンプに依存するMakefileは誤動作する恐れがあります。
- stat(2)が返すi-node番号はあてにはなりません。BOWでは各プロセス毎に64Kバイトのハッシュ表を持ち、そのハッシュ値を割り当てています。このハッシュは、"/"をstat(2)したとき、またはハッシュがあふれたときに初期化されます(getcwdが正常に動作するように作成されている)。
- オープン中のファイルのrename(2)はできません。このためmakedependは動きません。
- Windows 3.1の場合、uncompressなどa.outファイル名(argv[0])で動作を決定するコマンドはDOSファイル名の制限(8.3制限)で動作しない場合があります(uncompressの場合はcompress -dでuncompressできる)。

7.2.2 Windowsの仕様に関すること

- 32ビットコードで起こった一般保護例外は捕らえることは可能(「数値演算プロセッサ使用不可」は捕らえられない)ですが、EIP、ESPの上位16ビットが無効になってしまうため例外の発生したアドレスを特定できません。このため、trace(2)を実装することは不可能なので、gdbなどのデバッガは使用できません。したがってcoreファイルを作成してもあまり意味がないのでcoreダンプはしないようになっています。プロセス情報の詳細を知りたい場合は、BOWPSの詳細情報の表示機能を使ってください。
- 80x87が未実装のマシンで80x87のインストラクションが実行されるとマシンがフリーズします。Windowsプログラムの強制終了である、Ctrl + Alt + Deleteの入力も無効になります。
- Windows 3.1の場合、BSDプログラム(32ビットコード)を実行中にCtrl + Alt + Deleteでの強制終了は不可能です。実行するとWindowsが異常終了します。Windows 95では問題ありません。

7.2.3 ネットワークセキュリティに関すること

BOW はシングルユーザーモードを前提に設計されています。また、特権保護機能を前提に作られている BSD のネットワーク daemon をそのまま動作させることはネットワークに対するセキュリティホールになる危険性を秘めています。ネットワーク daemon は動作させないほうがよいでしょう。

BOW のシステムに telnet などでログインされた場合ファイルの保護ができないため不用意な削除をされる危険があります。

7.2.4 その他の制限

- BOW は BSD カーネルとはまったく異なる実装となっていてカーネルのデータ構造が異なるため、ps(1) などの/dev/kmem を読むプログラムは動きません。
- pty のパケットモードは実装していません。したがって、これを使っている screen、pty などのプログラムは動きません。
- マルチボリューム rfd(rfd?m.???)から圧縮されたファイルを読み出すと uncompress がオーバーランするためエラーメッセージがでますが実害はありません。
- IME を使用しないでカナ入力を行なうことはできません。
- シグナル割り込み後のシステムコールの再実行はサポートしていません。
- 2 バイトコードを含むファイル名には対応していません。
- manbow の変換は完全ではありません。独自に nroff のマクロを定義している ash のようなマニュアルは変換できません。

7.3 動作可能a.outの種類

BOW 上で動作する a.out の種類は次の通りです。複数の a.out に対応していますが混在して使用した場合の動作は保証できません。これはパスワードデータベースの形式が異なるためです。

OS	バージョン
4.4BSD-Lite	
FreeBSD	1.0、1.1、2.0

7.4　ファイルシステム

ここでは BOW に実装されているファイルシステムについて説明します。BSD と DOS のファイルシステムの違いへの対応や拡張などを解説しています。

7.4.1　Windows 3.1でのファイルシステム

BOW は DOS のファイルシステムを使用しています。そこで、BSD では DOS のファイルシステムでサポートされないファイル名を使用するために、次のようなファイル名のマッピングを行います。

BSD	DOS	例
,	’	main.c,v → MAIN.C’V
+	‘	g++ → G“
[	&	
名前の最初の〝.〟	$	.emacs → $emacs など
一番右以外の〝.〟	$	DOC-18.59.1 → DOC-18$5.1 など

これらの変換はファイルの open(2)、ディレクトリの read(2) 時に行なわれるので BSD で作成した tar などのアーカイブを BOW 上で解凍すれば自動的に変換されます。

また DOS の 8.3 制限を超えた部分は無視されます。逆にいえば、ファイルは 8.3 で区別できなければいけません（たとえば resource.h と resourcevar.h は同一のファイルとみなされる）。

またファイル名のアルファベットの大文字と小文字は区別されないので、たとえば Makefile と makefile は同一ファイルとみなされます。

BSD ファイルシステムでは 256 文字までの任意の英数字を使用できます。BOW 上では BSD の慣習に従って、次に示す文字はファイル名に使わない方がよいでしょう。

? @ # $ ^ & * ( ) ’ [ ] | ; ’ ” < > ! ~ ＼

また、ファイル名の先頭に〝+〟、〝-〟、〝.〟を付けるのも避けたほうがよいでしょう。これは BSD プログラムの初期化ファイルなどに使われているため、他のファイルに使った場合に混乱する可能性があるからです。

7.4.2 Windows 95でのファイルシステム

Windows 95 ではファイルシステムの仕様が拡張されたため、前項の制限はほとんど緩和されました。ただし、BSD のファイルシステムと比較して、次のような相違点があります。

- ファイル名中に大文字と小文字を混在させることができるが、ファイルへのアクセスの際には大文字小文字の違いは認識されない。このため、同一ディレクトリ中に、makefile、Makefile、MAKEFILE などのファイルを同時に作成することはできない。
- ファイル名の最後が"."（ピリオド）で終わるファイルは作成できない。"."で始まるファイル名は可能。
- ドライブ名、パス名、ファイル名を合計したファイル名の最大長は 259 文字まで。また、ファイル名単体での最大長は 255 文字まで。このため、ルートディレクトリには 255 文字までのファイル名をつけることができるが、サブディレクトリではドライブ名（「C:\」など 3 文字）とサブディレクトリ名（最低 1 文字＋区切り文字¥の 2 文字）が加わるため、最大でも 254 文字となる。さらに下の階層のファイルについては、上位ディレクトリのパス名の長さを 259 から引いた長さがファイル名の最大長になる。
- ファイル名には以下の文字は使用できない。

  " ＊ / : < = > ? ¥ |

7.4.3 ドライブファイルシステム（Drive File System）

BSD のファイルシステムは単一の木構造で構成されています。したがってドライブの概念を持っていない BSD のプログラムから DOS のドライブにアクセスするために、BOW では"A、B ... Z"ドライブを/dfs の下に"/dfs/a、/dfs/b ... /dfs/z"のようにマウントします（図 7-3）。

ネットワークの設定が行なわれていて、BOW Configuration ツールでホスト名の登録を行なっている場合には、"/dfs/[ホスト名]/[共有名]"の形式で直接ネットワークドライブを指定することも可能です。

このマウントは仮想的（単にファイル名の変換を行っているだけ）なので、マウント、アンマウントを行う必要はありません。メディアの交換は Windows 上で可能な時点ならばいつでも可能です。NetWare のドライブ、CD-ROM、MO なども使用できます。

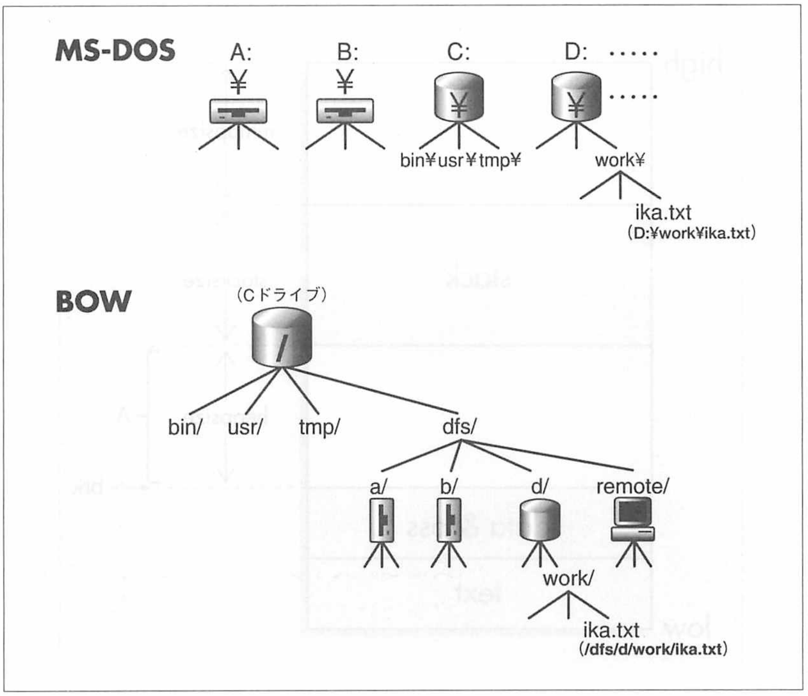


図 7-3　ドライブファイルシステム

またディレクトリのループを避けるために root ドライブはドライブファイルシステムの下にはマウントされません。

7.5 メモリマップ

BSD では自身でページ管理を行ない、ページが要不要になるたびに (brk 値が変更になるたびに) ページ割り当て、開放を行なうことができるので、A のスペースを広くとることができます (図 7-4)。メモリ空間だけを割り当ててページは必要時に割り当てます。

BOW では Windows が提供する仮想記憶をそのまま使用していますが、Windows API レベルではページの制御を行なうことができません。またスタック上のオブジェクトがポインタによって参照されている場合もあるため、スタックの領域を動的に移動することはできず自動拡張は不可能です。

このため適切な heapsize、stacksize をユーザーが設定する必要があります。しかし、最適なメモリサイズはプログラムの使用方法によって変化するため、一般的なサイズはなく、試行錯誤をしながら求めなくてはなりません。

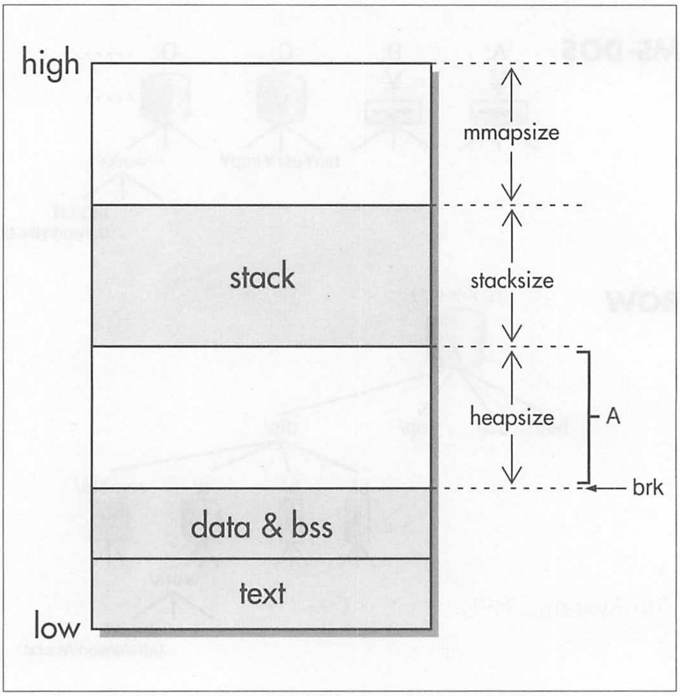


図 7-4　BOW のメモリマップ

7.6　ディレクトリ構成

BOW のディレクトリ構成は、BSD のディレクトリ構成をそのまま受け継いでいます。BSD 上で動作するプログラムのいくつかは、使用するファイルが特定のディレクトリにあることを前提にしているものがあります。

/bow(デフォルト)

　BOW のインストールディレクトリ。

/bow/samples

　elisp のサンプル、DOS プログラムの wrapper などが入っています。

/bow/unsuppor.ted

　manbow (nroff → rtf へのコンバータ)、doswrap.exe のソースが入っています。このディレクトリの下にあるプログラムはすべてパブリックドメインに置き、すべての権利と責任を放棄します。

/bin

/bin にあることが前提とされているコマンド (/bin/sh、/bin/pwd など) はここにインストールする必要があります。

/dev

デバイスファイルが作られます。DOS 上では大きさ 0 のファイルです。

/etc

passwd やネットワークの設定ファイルなどを置きます。

/usr/share/misc

termcap が置かれます。環境変数 TERMCAP を設定することで、termcap を配置するディレクトリを指定できます。

/usr/sbin、/usr/bin

vipw の実行には/usr/sbin/pwd_mkdb、/usr/bin/vi が必要です。

/dfs

Drive File System。BSD プログラムをインストールしたドライブ (ルートドライブ) 以外のドライブが、このディレクトリ以下にマウントされます。

WINDOWS ディレクトリ

BOW.INI および、そのバックアップファイルである BOWINI.OLD が置かれます。

windows/system

Microsoft 社提供のダイナミックリンクライブラリ CTL3D.DLL、WINMEM32.DLL が置かれます。

7.7 ドライバ

ここでは BOW の各種ドライバの仕様について説明します。BSD プログラムを作成する際に以降の説明を参考にしてください。

7.7.1 BOWPAD(コンソールドライバ)

BOWPAD は BOW のコンソール入出力を行う Windows プログラムです。ANSI 端末に近いエスケープシーケンス、BSD の termios による ioctl をサポート (4.3BSD の ioctl は termios ベースのものに libc が変換する) しているので termcap を利用した emacs などを使用することができます。termios 構造体の t_cflag の hupcl ビットは BOWPAD ウィンドウのオートクローズの制御に、t_iflag の IMAXBEL ビットは IME の制御に使用し

ています。

Windowsと連携する機能としてはドラッグアンドドロップに対応しています。ファイルがドロップされたときに(エスケープシーケンスで設定する)任意の文字列をキーボードからの入力として生成することができます。emacsにドラッグアンドドロップ機能を付加するelispコードがsamples/elisp/bowpad.elに入っているので、emacsのlisp/termにコピーしておけば、emacsにドロップすることができます(ただし、カレントバッファのキーバインドに依存するので完全ではない)。ドラッグアンドドロップはNON CANONICALモードでは利用できません。

フォントの指定はコントロールメニューで行います。これはWindows標準の[フォントの指定]ダイアログボックスを呼び出して行われます(「4.2.1 BOWPAD」の「フォントの選択」参照)。ウィンドウサイズ、フォントサイズの変更で表示文字数が変わったときにSIGWINCHがそのBOWPADを制御端末としているプロセスに送られます。SIGWINCHに対応しているemacsなどのプログラムは表示文字数が自動的に変更されます。またスクリーンサイズ変更をサポートするtelnetセッションのtelnetdにもサイズの変更が送られます(ただし、接続先のtelnetdが対応していなければならない)。

7.7.2 フロッピードライバ

/dev/rfd?[m].fmt (raw fllopy driver) tarなどで使用できます。rfd?m.???はマルチボリュームをサポートしていて、メディアの終端で交換のダイアログボックスを表示するようになっています(ただし、メディアの物理フォーマットはMS-DOSのデバイスドライバでサポートされていなければならない)。

BOW.INIの[config]セクションで指定するマイナーコード(10進数)を2進数にしたときの各ビットの意味は次の通りです。

mttddddd

m	multi volume
tt	drive type (BOW.INI BowRFDセクションのDiskParam[0-3]で指定)
ddddd	drive No.

7.7.3 Internet Domain Socketドライバ

ソケットドライバとして Windows Socket API 1.1(以降は WinSock と略す)に準拠した WINSOCK.DLL を利用しています。ソケットはデータグラム、ストリームの両方を使用できるので BSD のネットワークアプリケーションはそのまま利用可能です。

WinSock は local loopback について規定していないので、自分自身との通信は WINSOCK.DLL のインプリメントに依存します。

ソケットをプロセス間で共有するために一般のアプリケーションよりも WinSock を厳密に使用しています。そのため、WinSock の実装が確実でない WINSOCK.DLL ではうまく動作しない可能性があります。たとえば unblock なソケットコールでタスクスイッチを行う (GetMesssage を呼ぶ) WINSOCK.DLL は使用できません。

7.7.4 UNIX Domain Socketドライバ

ストリームソケットにのみ対応しています。使用できるソケットコールは socket、read、write、listen、accept、socketpair、connect、bind、close、fstat、fcntl です。emacs-client、server を使用することができます。BSD とは実装が異なり、bind を行ってもファイルシステムのネームスペースには socket はリンクされません。また、アクセス権の譲渡は実装されていません。

7.7.5 ptyドライバ

パケットモードは実装されていません。これを使用する screen は動作しません。

7.7.6 プリンタドライバ

コントロールパネルの[プリンタ設定]で指定されている、デフォルトのプリンタが接続しているポートへの RAW インターフェースを提供します。ドライバはコード変換などの加工を行なわないので、BSD 側のプログラムで書式設定などを行なう必要があります。

また、Microsoft ネットワークや NetWare などのネットワークプリンタも使用することができます。UNIX ホストに接続されているリモートプリンタ (lpd による通信を行なうもの) については、BOW 側では対応していません。

7.7.7　クリップボードドライバ

デバイスのマイナー番号でクリップボードのタイプを指定します。ドライバではデータを解釈していないので、BSDの側のプログラムで解釈する必要があります。emacsからテキストタイプのクリップボードを利用するコードがbowpad.elに入っています。

また、ライトモードでオープンされたクリップボードがクローズするときには「BowWin SyhchwClass」をクラス名に持つウィンドウにメッセージとしてGlobalAddAtom(”Bow WinSynchMessage”)の値を持つメッセージがポストされます。これによりWindowsアプリケーションがBSDプログラムとの同期をとることが可能になります。応用例は「4.1 WindowsおよびMS-DOSとの連携」を参照してください。

8章

プログラマーズガイド

この章では、BSD のシステムコールが BOW 上でどのように動作するかが記されています。特に注意が必要なソフトウェアに関しては、個別に説明をしています。また BOW 用に機能拡張されたソフトウェア、BOW 上では動作不可能なソフトウェアに関しても記しました。CD-ROM に収録されなかった数多くの BSD 用プログラムを BOW 上で動作させるときの参考にしてください。

8.1 BSDシステムコール

BOW では次の表に示すシステムコールをサポートしています。4.3BSD 以前でシステムコールとなっていたものもライブラリで提供されている場合があるので、この一覧に載っていないものでも使用できるものがあります (wait3、recv、send など) 。

システムコール	動 作	備 考
exit	○	
fork	○	
read	○	
write	○	
open	○	
close	○	
wait4	○	
unlink	−	オープン中のファイルはすぐには unlink されずに close 時に unlink されます。
chdir	○	
fchdir	○	
chmod	△	
chown	△	
break	○	
lseek	□	
getpid	○	
getuid	○	
geteuid	○	
recvfrom	○	
accept	○	
getpeername	○	
getsockname	○	
access	○	
syscall	○	
kill	○	
stat	□	
getppid	○	
lstat	□	
dup	○	
pipe	○	

システムコール	動　作	備　考
getegid	○	
sigaction	○	
getgid	○	
sigprocmask	○	
ioctl	○	
execve	＋	Windows および MS-DOS アプリケーションを実行できます。実行時の argv[0] が〝-″で始まる場合はアイコン化されます。また〝:″以降（ドライブ名を除いたパス名）は ShellExecute の第２引数として渡されます。registry database で DDE の指定がある場合はその指定で DDE コマンドを発行することができます。Windows API の ShellExecute の第２引数には〝open″、〝print″が指定できます。
umask	△	
fstat	□	
getpagesize		
vfork	−	vfork された子プロセスは親のプロセス空間をそのまま使用します。したがって vfork(2) のマニュアルに記述されているとおり、親の資源を変更してはいけません。これを守っていないプログラム（csh など）は BOW.INI の [debug] セクションの VforkDataCopy を yes にすることにより救済できます。この指定を行うとスピードが低下します。
mmap	□＋	新しい mmap は 4.4BSD-Lite の仕様、古い mmap は Free BSD 1.1 の仕様。
	−	PROT_READ の mmap は mmap 時にファイルの読み込みを行う事でエミュレートしています。
unmmap	−	PROT_WRITE の mmap は unmmap 時にファイルの書き出しを行う事でエミュレートしています。
mprotect	△	
getpgrp	○	
setpgid	○	
setitimer	○	
gethostname	○	
getdtablesize	○	
dup2	○	
fcntl	○	
select	○	
fsync	○	
socket	○	
connect	○	
sigreturn	○	

システムコール	動　作	備　考
bind	○	
setsockopt	○	
listen	○	
sigsuspend	○	
gettimeofday	○	
getsockopt	○	
getrusage	○	
writev	○	
fchmod	○	
setreuid	○	
setregid	○	
rename	−	オープン中のファイルの rename はできません。
flock	○	
sendto	○	
shutdown	○	
socketpair	○	
mkdir	○	
truncate	□	
ftruncate	□	
utimes	−	ディレクトリのタイムスタンプは変更できません。
rmdir	○	
getrlimit	□	
setrlimit	□△	
setsid	○	
getdirentries	○	4.4BSD-Lite と FreeBSD 1.1 の仕様の両方をサポート。
statfs	○	
fstatfs	○	
getdomainname	○	
setdomainname	○	
uname	○	
sysctl	○	

○　未修整のシステムコール
−　386BSD に対し制限のあるシステムコール
△　何もしないでリターンするシステムコール
＋　拡張されているシステムコール
□　32bit オフセットの古いシステムコールと 4.4BSD-Lite で採用された 64bit オフセットの新しいシステムコールの両方サポート

8.2　フリーソフトウェアのインストール

BOWはBSDに比べてファイル名の制限が厳しくなっています。そのため、BOW上でプログラムのコンパイルを行う場合には、Makefileなどの書き換えが必要です。DOSのファイル名に当てはまらないものは注意が必要です。ここでは対処法の一例を示しますが、これがすべてではありません。なお、Windows 95上でBOWを使用する場合は、ファイル名の制限による不具合は解消します。ファイル名の制限に関連する記述については、Windows 3.1で使用する場合に固有のものと考えてください。

8.2.1　bison

ファイル名の制限によるmakeの誤動作を避けるために、makefileに次の依存関係を記述します。

```
LR0.o: LR0.c
conflicts.o: conflicts.c
```

8.2.2　GCC

stamp-*、tmp-*がファイル名の制限により重複します。makefileのstamp-をst-に、tmp-をtm-に書き換えます。

8.3　BOW用に拡張されたソフトウェア

ここではBOWで利用するために拡張したものや変更をしたものを紹介します。これ以外のBSDプログラムをBOWように拡張する際の参考にしてください。

8.3.1　emacs

emacsは無修正で動作することが可能ですが、Windows上でより使い易くするために、Windows向きに拡張、修正されています。

1. プログラムマネージャから直接起動されたときに、suspendを無効にする。
2. クリップボードからのコピーを次のコマンドで拡張しています。

bowpad-yank-clipboard-contents
クリップボードの内容を挿入する

3. クリップボードへのコピーを次のコマンドで拡張しています。

bowpad-copy-region
mark と point の間の領域をクリップボードにコピーする

4. ファイルマネージャからファイルをドラッグアンドドロップすることができます。ドロップ時に、「C-xC-f ファイル名 C-m」がキーボード入力として送られます。
5. バックアップファイル名がファイル名の制限に該当しないようにしています(Windows 3.1 の場合)。
6. IME の状態 (変換モードが ON か OFF か) を emacs から検知できるようにしています。また、emacs から IME の ON/OFF を制御することも可能です。この機能を使う場合は、BOWPAD のコントロールメニューの「IME Notify」をチェックします。

8.3.2 exec.c

本来 vfork されたプロセスは execve するまでは親プロセスとデータ領域を共有するためデータ領域の変更を行なってはならないことになっていますが、オリジナルの execl などのライブラリ中で malloc を呼び出しています。このため BOW 上での実行に不向きなので、malloc を使用しないように修正しています。

8.4 動作不可能なプログラム

ここでは BOW の仕様により動作が不可能なプログラムについて説明します。

- **カーネルメモリを読むプログラム**：ps や netstat のようにカーネルのデータ構造に依存するプログラムは実行できません。ps については Windows ベースの BOWPS を用意しています (「4.2.3 BOWPS.EXE」を参照)。
- **root の特権が必要なプログラム**

APPENDIX

BSD *on* Windows

APPENDIX A
BOWPAD エスケープシーケンス

BOWPADで利用できるエスケープシーケンスを示します。行頭に「○」が付いているものはBOW用に拡張されたものです。また、「WindowText」などはコントロールパネルで設定するWindowsの画面要素を表しています。

	エスケープシーケンス	動　作
	ESC [*pl*;*pc*H	カーソルを*pl*行*pc*カラムに移動させる。
	ESC [*pl*;*pc*f	カーソルを*pl*行*pc*カラムに移動させる。
	ESC [*pn*A	カーソルを*pn*行上の同一カラムに移動させる。
	ESC [*pn*B	カーソルを*pn*行下の同一カラムに移動させる。
	ESC [*pn*C	カーソルを*pn*文字右に移動させる。
	ESC [*pn*D	カーソルを*pn*文字左に移動させる。
	ESC [0J	カーソル位置から最終行の右端までをクリアする。
	ESC [1J	先頭行の左端からカーソル位置までをクリアする。
	ESC [2J	画面全体をクリアし、カーソルをホーム位置へ移動させる。
	ESC [0K	カーソル位置から行の右端までをクリアする。
	ESC [1K	行の左端からカーソル位置までをクリアする。
	ESC [2K	カーソルが位置する行の左端から右端までをクリアする。
	ESC [*pn*M	カーソルが位置する行から下を*pn*行削除する。
	ESC [*pn*L	カーソルが位置する行の上に*pn*行の空白行を挿入する。
	ESC [*ps*;*pe*r	*ps*行から*pe*行迄をスクロール領域として設定する。
	ESC D	カラム位置をそのままに、カーソルを一行下に移動させる。カーソルがボトムマージンにある場合は、一行スクロールアップする。
	ESC E	カーソルを一行下の左端に移動せさる。カーソルがボトムマージンにある場合は一行スクロールアップする。
	ESC M	カラム位置をそのままに、カーソルを一行上の行に移動させる。カーソルがトップマージンにある場合はスクロールダウンする。
○	ESC W	termcapモードに入る
○	ESC X	termcapモードから抜ける。
	ESC *	画面全体をクリアし、カーソルをホーム位置へ移動させる。
○	ESC ! *xfmtx*	任意の文字*x*で挟まれた文字列*fmt*をドラッグアンドドロップ時のwsprintfのフォーマットとする。*fmt*が空の場合はドラッグアンドドロップが禁止される。wspritnf(buf,fmt, ファイル名)で生成される文字列がキー入力としてプロセスに渡される。

エスケープシーケンス	動　作
ESC [6n	カーソル位置の取得。
ESC [*ps*;...;*ps*m	表示文字の属性を設定する。*ps*の値については次の表を参照すること。

ps の設定値

ps の値	属　性	デフォルトの画面要素	BOW Configuration での色番号
0	規定値		-
4	アンダーライン		-
7	リバース		-
30	フォアグランド	WindowText	0
31	〃	CaptionText	1
32	〃	MenuText	2
33	〃	HighlightText	3
34	〃	Highlight	4
35	〃	Menu	5
36	〃	ActiveCaption	6
37	〃	Window	7
40	バックグランド	WindowText	0
41	〃	CaptionText	1
42	〃	MenuText	2
43	〃	HighlightText	3
44	〃	Highlight	4
45	〃	Menu	5
46	〃	ActiveCaption	6
47	〃	Window	7

APPENDIX B キーテーブル

CANONICALモード、NON CANONICALモードではコード変換を行ないます。DLL(JCONV.DLL、NOCONV.DLL)のアクセラレータによって機能コードが発生します。

■CANONICALモード

スクロールや画面上の文字の入力エリアへのコピーなどの機能が使用でき、それらの機能は以下のようにキーに割り当てられています。入力エリアでの文字編集も可能です。

Scroll Lock	表示エリアのスクロールの停止／再開を切り替えます。BOWPADのタイトルバーに現在の状態が表示されます。PC-9800シリーズでは vf・5 キー。
Insert	カーソル以降の文字を入力エリアにコピーします。
↑	カーソルの移動(表示エリア)
↓	〃 (〃)
←	〃 (表示エリアと入力エリア)
→	〃 (〃)
Page Up	スクロールアップ
Page Down	スクロールダウン
Home	カーソルを行頭に(表示エリアと入力エリア)
End	カーソルを行末に(表示エリアと入力エリア)
Delete	カーソル位置の1文字を消去(入力エリア)
Esc	カーソルを入力エリアに戻す(表示エリア)

■ NON CANONICAL

NON CANONICAL（CBREAK および RAW）モードでは、行編集機能は BSD プログラムの機能に依存します。BOWPAD では、以下の表に定義するアクセラレータについては、特定のキャラクターコードを割り当てています。これは、emacs などのテキストエディタを使用する際の便宜を図るものです。

アクセラレータ	キャラクターコード
Home	Ctrl+A
Alt − Home	Meta−<
End	Ctrl+E
Alt − End	Meta−>
Page Up	Meta−V
Page Down	Ctrl+V
↑	Ctrl+P
↓	Ctrl+N
←	Ctrl+B
→	Ctrl+F
Delete	Ctrl+D
BS	DEL
Ctrl + Space	Ctrl+@
Alt + Ctrl + Space	Meta−Ctrl+@
Ctrl + ?	Ctrl+_
Ctrl + A から Z	Ctrl+A から Z
Alt + Ctrl + A から Z	Meta−Ctrl+A から Z
Alt + Ctrl + a から z	Meta−Ctrl+A から Z
Alt + a から z	Meta−a から z
Alt + A から Z	Meta−A から Z

APPENDIX C
FAQ

Q1. tcsh (bash) で画面のバックスクロールができません

A1. tcsh (bash) はコマンド入力時に端末を CBREAK モードに設定するため、バックスクロールはできません。一時的に CANONICAL モードにすることにより可能になります。tcsh では shell 変数 edit を unset することで CANONICAL モードにできます。

Q2. telnet で SUN/OS 4.1 にログインしたときにウィンドウサイズが正しく設定されません

A2. SUN の telnetd の仕様です。telnetd を画面サイズ変更に対応した物に取り替えるか、X11 のリリースに含まれている resize をウィンドウサイズの変更の際に使用してください。

Q3. mule、vi で IME が使用できません

A3. BowPad のメニューで漢字コードを正しく設定してください。

Q4. vi でカーソルキーが使用できません

A4. 使用できません。emacs を使いましょう。

Q5. BOW の動作原理を教えてください

A5. DPMI のファンクション 0DH (alloc spicific selector)、WINMEM32.DLL、公開されている Windows 3.1 の API だけで動いています。VxD (仮想デバイスドライバ) は使用していません。

Q6. BOWPS で BOW から起動した DOS アプリケーション、Windows アプリケーションが kill できません

A6. そのプロセスは BOW から起動した DOS アプリケーションまたは Windows アプリケーションの終了の同期をとるために存在している擬似的なプロセスなので kill できません。BOW の DOS アプリケーション、Windows アプリケーションの起動は PROGMAN.EXE を置き換えるようには設計されていません。

Q7.　ls で日本語のファイル名が正しく表示されません

A7.　ls の仕様です。出力先が端末のときは 8bit スルーにはなりません。ls | crpad などで、日本語のファイル名を表示することは可能です。

Q8.　XFree86 は動きますか?

A8.　XFree86 は動きません。

Q9.　ftpd、telnetd が動きません

A9.　BSD の ftpd、telnetd をそのまま使うことはセキュリティ上の重大な問題があるため、ftpd、telnetd などで必要な setuid などのシステムコールはサポートしていません。winsock 対応の ftpd、telnetd などを使用してください。

Q10.　execvp - foo.doc foo.doc で起動した MS Word を終了するとハングアップします

A10.　そのように MS Word が作られているのが原因です。MS Word の文章を「execvp -」で起動したい場合は Windows に付属している REGEDIT.EXE で、DDE を使用しないでコマンドラインで指定する方法に変更してください。

Q11.　Vz Editor を常駐させたまま Windows を起動すると BOW が起動できません

A11.　コンベンショナルメモリが足りない可能性があります。Vz Editor を常駐させないかコンベンショナルメモリが増えるようにその他の常駐プログラムなどの設定を変更してください。

Q12.　「Can't load library.」というメッセージが出て、BOW のインストールや起動ができません

A12.　Q11 と同じく、コンベンショナルメモリの不足が原因です。Windows 3.1 で BOW を使う場合によく発生するエラーです。不要なデバイスドライバや TSR（常駐型プログラム）の使用を中止して、コンベンショナルメモリの空き容量を約 520K バイト以上確保してください。BOW 1.5 では Windows 95 での利用を推奨しています。

APPENDIX D 参考文献

■UNIXの入門書

・入門UNIX

羽山 博　著

アスキー出版局

続刊の「実用UNIX」、「応用UNIX」へとレベルアップしていけるテキスト。

・たのしいUNIX － UNIXへの招待

坂本 文　著

アスキー出版局

いますぐUNIXを使いたい人のための本。BOWにもってこいの1冊です。

・続・たのしいUNIX － シェルへの招待

坂本 文　著

アスキー出版局

インストールで、「/bin/csh」を選んだ人はぜひどうぞ。

■環境の設定

・UNIXの環境設定

久野 禎子／久野 靖　共著

アスキー出版局

UNIXの環境設定の入門書。.login、.cshrc、.profileなどの設定などに役立ちます。

・termcap & terminfo

John Strang／Linda Mui／Tim O'Reilly　共著

鈴木 麗　監訳

アスキー出版局

端末の設定に欠かせない1冊です。

■ プログラミングのための本

- 改訂第 3 版 C 言語入門

  Les Hancock／ Morris Kriger／ Saba Zamir　共著

  倉骨 彰／三浦 明美　共訳

  アスキー出版局

  UNIX のプログラミングに不可欠な C 言語の入門書です。

- UNIX プログラミング環境

  Brian W.Kernighan／Rob Pike　共著

  石田 晴久　監訳

  アスキー出版局

  UNIX の基礎的なプログラミングを解説しています。

- UNIX C プログラミング

  David A. Curry　著

  アスキー書籍出版部　監訳

  アスキー出版局

  UNIX の機能を活かした上級プログラミングの解説書です。

- Perl 書法

  増井 俊之　著

  アスキー出版局

  プログラミングツールとしての Perl の実際的なテクニックの紹介。

- lex & yacc プログラミング

  John R. Levine／Tony Mason／Doug Brown　共著

  村上 列　訳

  アスキー出版局

- プログラミング言語 C 第 2 版

  Brian W. Kernighan／Dennis M. Ritchie　共著

  石田 晴久　訳

  共立出版

  「hello, world」で有名な C の原典。通称、「K & R」。

■ UNIX プログラムの解説

・sed & awk プログラミング

Dale Dougherty　著

福崎 俊博　訳

アスキー出版局

付属の CD-ROM にも収録されているフィルタツールの入門書です。

・入門 Mule

大木 敦雄　著

アスキー出版局

付属の CD-ROM にも収録されているエディタの入門書です。

・入門 Perl

河野 真治　著

アスキー出版局

付属の CD-ROM にも収録されているプログラミングツールの入門書です。

・vi 入門

Linda Lamb　著

福崎 俊博　訳

アスキー出版局

BSD を使う上で必要不可欠な vi エディタの入門書です。

■ BSD や UNIX の詳細を知るための本

・プロフェショナル BSD

砂原 秀樹／石井 秀治／植原 啓介／林 周志　共著

アスキー出版局

「BSD を知るための 1 冊」です。

・UNIX セキュリティ

Simson Garfinkel／ Gene Spafford　共著

山口 英　監訳

アスキー出版局

・UNIX システムチューニング

Mike Loukides　著

砂原 秀樹　監訳

アスキー出版局

・BSD を 256 倍使うための本

宮川　晋／増田佳泰／古場正行　共著

アスキー出版局

BSD システムの歴史や仕組みについて、親しみやすい語り口で解説しています。

・UNIX を愛するすべての人に Life with UNIX

Don Libes／Sandy Ressler　共著

坂本 文　監訳

福崎 俊博　訳

アスキー出版局

UNIX の歴史動向、人名辞典、アングラ情報などを網羅しています。

索　引

Symbols

A

B

C

D

E

F

G

H

I

■著者紹介

太田博志

神奈川県出身。大手電機メーカーでLSI CADの開発に従事。快適なプログラミング環境を構築することに興味を持っている。最近はOLEに興味を示しつつあり、muleとの融合に期待を抱いている。「Win32は32ビットしか使えないが、Win16は48ビット使える」というのが、BOWのWin32化が遅れている言い訳とのこと。

■CD-ROM編集協力

富樫秀昭

■パッケージデザイン

Mike Smith (DO THE MONKEY)

■アイコンデザイン

森田福助

■編集協力

永田淳一

●本書の内容に関するご質問は、小社スーパーアスキー編集部まで、封書（返信用切手同封のこと）にてお願い致します。
電話によるお問い合わせには、応じられません。
なお、本書の範囲を越える質問に関しては、お答えできない場合もあります。
●落丁・乱丁本は、送料当社負担にてお取り替え致します。
お手数ですが、小社営業部までご返送ください。

Matrox未対応（96.10.8現在）
http://www.ascii.co.jp/pb/superascii/bow/index.html

BSD on Windows Version 1.5

1996年5月11日　初版発行
1996年6月11日　第1版第2刷発行

著　者　太田 博志（おおた ひろし）
発行人　宮崎 秀規
編集人　土屋 信明
発行所　株式会社アスキー
〒151-24　東京都渋谷区代々木4-33-10
振　替　00140-7-161144
大代表　(03)5351-8111
出版営業部 (03)5351-8194（ダイヤルイン）
スーパーアスキー編集部 (03)5351-8125（ダイヤルイン）

制　作　株式会社GARO
印　刷　凸版印刷株式会社

編　集　竹田 善太郎／佐々木 三奈

ISBN4-7561-1903-4　　Printed in Japan

●1322189

返 却 期 限 票

- ○ 最後にある日付があなたの返却期限です。
- ○ 遅れないように期限内に返却しましょう。
- ○ 続いて借りたいときは届け出てください。

'96.11.26		

BSD on Windows 1.5

BSD on Windows 1.5は、Windows 3.1あるいはWindows 95上で、4.4BSD-Liteベースのバイナリプログラムを動作させる、シングルユーザー、マルチタスクのBSDカーネルエミュレータです。Windows上に、テキストベースのUNIX系シェル環境を構築できます。bash、csh、tcshなどのシェル、mule、viなどの高機能テキストエディタ、perl、sed、awkなどの高機能フィルタツール、 gcc、as、cpp、makeなどのプログラミング環境が使えます。Windows 95上では、従来のMS-DOS/Windows 3.1に存在したファイル名の制限が、大幅に緩和されました。また、Windows 95のネットワーク機能を、BSDプログラムから利用することも可能です。

●パッケージ内容

・BOW 1.5ユーザーズマニュアル 1冊
・BOW 1.5システムフロッピー(3.5inch/1.44Mバイト/2HD) 1枚
・BSDプログラムCD-ROM 1枚

●必要なハードウェア構成

・対象機種
i386以上のインテル製CPU(あるいは相当品)を搭載し、Windows 3.1あるいはWindows 95が正常に動作するコンピュータ。i386やi486SX/SLなどのFPU機能を内蔵していないCPUを使用する機種については、別途FPU(i387など)やFPU機能をもつODPなどの追加が必要

・必要メモリ
Windows 3.1あるいはWindows 95が正常に動作するメモリ容量が必要。推奨16Mバイト以上。Windows 3.1で使用する場合は、コンベンショナルメモリに520Kバイト以上の空き容量が必要

・フロッピーディスクドライブ
1.44Mバイト/2HD形式の3.5インチフロッピーディスクを読むことのできるドライブが必要

・ハードディスクドライブ
MS-DOSあるいはWindows 95でフォーマットされた、以下の空き容量をもつハードディスクドライブのパーティションが必要。FreeBSD、NetBSD、BSD/386、LinuxなどのPC-UNIXでフォーマットされたドライブやパーティションへはアクセス不可。MS-DOSやWindowsとのファイル共有は可能
最低5Mバイト(BSDコマンドをCD-ROM上から利用する場合)
最大80Mバイト(すべてのBSDコマンドをハードディスク上にインストールする場合)

・CD-ROMドライブ
Windows 3.1あるいはWindows 95からアクセスできるドライブ。Windows 95のネットワーク機能を利用してアクセスできるものでも可

●必要なソフトウェア構成

・Windows 3.1あるいはWindows 95(Windows 95を推奨。Windows 3.1の場合、いくつかの機能に制限があります)。Windows/NT上では一切動作しません
・Windows 3.1でネットワーク機能を利用する場合、Windows Sockets API 1.1以上に準拠したネットワークドライバが必要。Windows 95の場合は、Windows 95に付属のTCP/IPドライバを利用可能

●動作可能なBSDバイナリの種類

・インテルCPU用の4.4BSD-Liteベースの実行バイナリファイルで、ANSI互換のキャラクタ端末の環境で動作するもの。X WindowなどのGUI環境を必要とするものは動作しません。また、gdbなどのデバッグツールは動作しません

●「BSDプログラムCD-ROM」収録ソフトウェア

・4.4BSD-Liteの全ソースファイル・/bin、/usr/bin、/usr/local/binなどのコマンドとそのソースファイル
・各種GNUプログラム(mule2、gccなど)およびそのソースファイルとBOWでコンパイルするための差分ファイル
・アスキー日本語TeX(pTeX 2.1、pLatex2e)

◎ご注意

・本製品は書籍として販売されるものです バグ修正などに伴う差分ファイルの公開を除き、将来のバージョンアップに伴う有償・無償のバージョンアップ サービスは一切行ないません また、電話による質問受け付けやユーザーサポートも行なっておりませんのでご注意ください

・本製品中のフロッピーディスク(「BOW 1.5システムフロッピー」)をパッケージから取り出す前に、フロッピーディスクを収めたパッケージに印刷されている「エンドユーザー使用許諾契約書」をよくお読みになり、すべての条項に同意される場合のみ、フロッピーディスクを開封してください フロッピーディスクを取り出した後での返品や交換は、フロッピーディスクもしくはCD-ROMの媒体の不良、マニュアルの乱丁・落丁、パッケージ内容物のご購入時点での破損や不足の場合を除き、一切受け付けませんのでご了承ください

・本製品中のCD-ROM(「BSDプログラムCD-ROM」)に収録したフリーソフトウェアの使用と再配布ついて、(株)アスキーは一切の制限を設けません ただし、使用や再配布に際しては、各ソフトウェアに付属するドキュメントファイルなどに記述された、ソフトウェア作者や一次配布元の指示に従ってください 本製品中のCD-ROMに収録したプログラムについてのサポートは、(株)アスキーでは一切行ないません

※フリーソフトウェアの健全な発展を期すために、本製品1部につき0.5米ドル相当の金額を、(株)アスキーが責任をもって米国Free Software Foundationに寄付します
We will donate half US dollar or equivalent to the Free Software Foundation if you purchase this package.